TAMPERE UNIVERSITY OF TECHNOLOGY

JUSSI AHOLA

PHOTOVOLTAIC RESEARCH POWER PLANT MEASURING AND DATA STORING SYSTEM

Master of Science Thesis

Examiner: Professor
Seppo Valkealahti
Examiner and topic approved in the
faculty of Computing and Electrical
Engineering Council meeting on
6th April 2011

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Programme in Electrical Engineering
AHOLA, JUSSI JANNE KRISTER: Photovoltaic research power plant measuring and data storing system
Master of Science Thesis, 78 pages, 3 Appendix pages
April 2013
Major: Electrical energy
Examiner: Professor Seppo Valkealahti
Keywords: Photovoltaic power plant, SQL, Database, LabVIEW, MATLAB

The photovoltaic research power plant of the Department of Electrical Energy Engineering at the Technical University of Tampere consists of 69 solar modules in different configurations, 21 solar (SP Lite 2) and temperature (Pt100) sensors, data measuring and logging cards (cRIO 9074, NI 9144, 2 NI 9205 and 6 NI 9217) and two weather stations with temperature and humidity sensors (HMP155), wind sensor (WS425) and irradiance sensors (CMP21 and CMP22).

This thesis is about connecting these devices together and creating an easy to use and efficient but complex system which processes, collects, saves and provides data from the sensors to researchers. The main goal is to provide researchers with an easy access to the data from the sensors from extended time periods. In the thesis different options and methods to implement the parts of the system have been studied carefully before implementing.

The main parts of the system implementation are the following: The data from the sensors are transmitted via cables and extension cables to the data measuring and logging cards which are connected to the local network. Sensors are either grounded from the cable or from the structure. In addition, a lightning rod was built and installed to protect the wind sensor and personnel. The data processing, collecting and saving implementations have mostly been done in LabVIEW. The data are processed by averaging the data to reduce noise and sending it at a 10 Hz rate to the host computer to be saved into a PostgrSQL database. A file-based system alternative to store the data was also implemented. A graphical user interface (GUI) implementation to the MATLAB was made to help researchers gain specific data from the database more easily. A public Internet site was created in the end for the system, and it offers limited general information to everyone who is interested.

The system was successfully implemented and completed. It is running 24/7 and ready for researchers to take advantage. Everything in the system has been made with the focus that the system will be expanded, and it should be easy to do so in future. It is recommended to study and understand how the system works before expanding it.

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO
Sähkötekniikan koulutusohjelma
AHOLA, JUSSI JANNE KRISTER: Aurinkosähkön tutkimusvoimalan mittaus- ja tiedontallennusjärjestelmä
Diplomityö, 78 sivua, 3 liitesivua
Huhtikuu 2013
Pääaine: Sähköenergia
Tarkastaja: professori Seppo Valkealahti
Avainsanat: aurinkosähkövoimala, SQL, Tietokanta, LabVIEW, MATLAB

Tampereen teknillisen yliopiston sähkötekniikan laitoksen aurinkosähkön tutkimusvoimala sisältää 69 aurinkopaneelia eri kytkennöillä, 21 säteily- (SP Lite 2) ja lämpötilasensoria (Pt100), tiedon mittaus- ja keräyskortit (cRIO 9074, NI 9144, 2 NI 9205 ja 6 NI 9217) ja kaksi sääasemaa, jotka koostuvat lämpötila- ja kosteussensoreista (HMP155), tuulisensorista (WS425) ja säteilynvoimakkuussensoreista (CMP21 ja CMP 22).

Tämä opinnäytetyö kertoo näiden laitteiden yhdistämisestä ja monitahoisen mutta helppokäyttöisen mittausjärjestelmän luomisesta. Kyseinen järjestelmä käsittelee, kerää, tallentaa ja tarjoaa mittalaitesensoreiden keräämän tiedon tutkijoille. Päätavoite on tarjota tutkijoille helppo pääsy tietoon pitkiltä ajanjaksoilta. Tässä opinnäytetyössä eri vaihtoehtoja ja tapoja toteuttaa järjestelmän osia on tutkittu tarkasti ennen niiden toteuttamista.

Järjestelmässä tieto kuljetetaan sensoreilta kaapeleilla ja jatkokaapeleilla tiedon mittaus- ja keräyskorteille, jotka ovat yhdistettyinä lähiverkkoon. Sensorit on maadoitettu joko maadoitusjohdosta tai rungostaan. Lisäksi rakennettiin ukkosenjohdatin suojaamaan tuulisensoria ja henkilöstöä. Tiedon prosessointi, keräys ja tallennus on toteutettu pääasiassa LabVIEW:llä. Tiedot prosessoidaan keskiarvoistamalla, jotta kohina poistuu. Tämän jälkeen ne lähetetään 10 Hz nopeudella palvelinkoneelle, jossa ne tallennetaan PostgreSQL-tietokantaan. Järjestelmälle luotiin myös vaihtoehtoinen tiedostopohjainen tallennusjärjestelmä. MATLAB-ohjelmaan luotiin graafinen käyttöliittymä helpottamaan tutkijoita saamaan tietty tieto tietokannasta. Julkinen internetsivu luotiin järjestelmän toteutuksen loppuvaiheessa, ja se tarjoaa rajoitettua yleistä tietoa kaikille kiinnostuneille.

Järjestelmä luotiin ja toteutettiin onnistuneesti. Se on jatkuvatoiminen ja on valmis tutkijoiden hyödynnettäväksi. Järjestelmä on toteutettu siten, että sitä voidaan laajentaa helposti tulevaisuudessa. On suositeltavaa tutkia ja ymmärtää järjestelmän toiminta ennen laajentamista.

PREFACE

At the very beginning when I saw the opportunity to do this kind of thesis, I knew I would enjoy doing it. When I read the description and noticed the requirements matching my studies and skills very well, I was certain that I would have a great opportunity to learn something new and use my studies and skills in practice.

Making this thesis has been an interesting journey. It has included events such as exchange studies in Canada and cooking a graphic card in the oven to get the computer used to write the thesis to run again.

I would like to thank Suvi Varttinen for the support with English, as well as Tommi Keikko and Seppo Valkealahti for the support and guidance I have got for this thesis. I am also very grateful for the help I have got from my friends and colleagues.

Tampere, 8th March 2013

Jussi Ahola

CONTENTS

SYMBOLS AND ABBREVIATIONS

B	The latitude of the observation site
C	Correction factor
D	Sun's declination
R	Radius
S	The intercepted part of the downward component of the sky radiation
U	The hour angle
U_0	The angle between the sun at sunrise (or at sunset) and the sun at true noon in the plane of the shadow ring CM121C
V	The ring subtending a solid angle from the center of the ring
Z	Zenith angle

a-Si	Amorphous silicon
AC	Alternating current
AM0	Air Mass 0 (sunlight spectrum before it passes the Earth's atmosphere)
AM1.5	Air Mass 1.5 (sunlight spectrum that arrives to the Earth's surface which represents the sun 48 ° overhead)
ANSI	American National Standards Institute
API	Application programming interface
CdTe	Cadmium telluride
CIS	Copper indium selenide
CSS	Cascading Style Sheet
DBMS	Database Management System
DC	Direct current
DEEE	Department of Electrical Energy Engineering

FPGA	Field-programmable gate array
FTP	File Transfer Protocol
GNU	GNU's Not Unix, Linux OS
GUI	Graphical user interface
GUIDE	Graphical user interface development environment
HTML	HyperText Markup Language
I/O	Input/output
I-V	Current-voltage
IIS	Internet Information Services
IrDA	Infrared Data Association
ISO	International Organization for Standardization
JDBC	Java Database Connectivity
JIT	Just-In-Time
LabVIEW	Laboratory Virtual Instrumentation Engineering Workbench
LED	Light-emitting diode
LiCl	Lithium chloride
MATLAB	Matrix laboratory
MAX	Measurement & Automation Explorer
MIT	Massachusetts Institute of Technology
MPPT	Maximum power point tracking
NaCl	Sodium chloride
ODBC	Open Database Connectivity
PHP	PHP: Hypertext Preprocessor
PV	Photovoltaic
RDBMS	Relational database management system

RH	Relative humidity
RTD	Resistance temperature detector
Si	Silicon
SQL	Structured Query Language
TUT	Tampere University of Technology
TCP/IP	Transmission Control Protocol/Internet Protocol
UDP	User Datagram Protocol
UI	User interface
UPS	Uninterruptible power supply
URL	Uniform resource locator
UV	Ultraviolet
VI	Virtual Instrument
VISA	Virtual Instrument Software Architecture
Win64	64-bit version of Windows

1. INTRODUCTION

Today people want to decrease the use of fossil resources and decrease the production of atmospheric emissions. This is because of the problems all this waste can cause if we let it pile in our world. Using renewable energy sources is a possible way to reduce this waste. Solar energy has high potential of being an excellent renewable energy source as it can be easily exploited nowadays.

Photovoltaic research power plant at the Department of Electrical Energy Engineering of Tampere University of Technology consists of 69 solar modules in total. In addition to this, several solar, temperature and other weather sensors had been ordered for creating a measuring system for the photovoltaic research power plant. To support these sensors, data logging devices had also been ordered.

This thesis is about a photovoltaic research power plant's data acquisition, preparing, storing and providing data for later research use. The main goal is to archive and provide the power plant data accurately so that it contains information from extended time periods and will thus be beneficial for multiple research topics. This is not a usual research thesis where something new is discovered but a thesis where a system has been built using existing technologies. The different technologies and choices have been carefully studied before implementing in the system.

In order to make this system fully viable, efficient and easily usable, several tasks had to be carefully designed and implemented. The first task will be connecting the devices together and grounding them. Secondly, the data from signal will have to be verified and filtered from possible noise. Thirdly, the filtered data will have to be transferred from the data logger into the host computer. Fourthly, the data will have to be saved in the best possible way while keeping in mind the future usage. This system will have to be easily usable and extendable as it is expected that the system will grow in the future. Lastly the presentation of the data has to be so that researchers and other users can easily get what they want within reasonable time and effort. All this has to be secure and take into account the possible errors that the system can encounter.

Because of the requirements, the system will be in real time and all the possible real time challenges must be solved. This will provide complicated challenges for the system implementation.

The present thesis begins with a brief theory part presenting the photovoltaic power plant, solar energy, distributed systems and database systems. After that, the department's photovoltaic research power plant layout and the main components will be described. The working conditions, specs and calibration data of each component will also be listed at this stage. The following chapter deals with the actual implementation of the system. This includes connections, grounding, different data storing systems, analyzing tool, public promotion and the main coding implementations in LabVIEW. Finally, the results of the project conclude the thesis along with appendixes.

2. THEORY AND BACKGROUND

The theoretical part of this thesis mostly consists of discussion on photovoltaic phenomenon, photovoltaic power plants, data warehousing and relational databases. In addition, some less significant and more specific theoretical aspects will be referred to when seen useful. Photovoltaic phenomenon and photovoltaic power plants will only be described briefly because they are not the main focus of this thesis. Data warehousing, on the other hand, will be discussed in more detail because of being of great importance in this thesis.

2.1. Photovoltaic phenomenon

In technological meaning, photovoltaics is what generates direct current electrical power and it comes from semiconductors when illuminated by photons. Electrical power is generated as long as light is shining on the solar cell, and naturally, when there is no light, no electrical power is either generated. The information in this section is modified after Handbook of Photovoltaic Science and Engineering by A. Luque and S. Hegedus. [1]

Photovoltaics has it advantages and disadvantages. Usually these advantages and disadvantages are completely opposite to those of conventional fossil-fuel power plants. What they do have in common is that both fossil fueled and photovoltaic power plants are reliable. The major photovoltaic advantages and disadvantages are listed in Table 2.1.

Photovoltaic physical basis is today well defined. Semiconductors are used as the base material for solar cells. They have weakly bonded electrons in a band of energy called the valence band. In the conduction band the electron can conduct electricity through the material. Thus, by the band gap (measured in units of electron volts) the free electrons in the conduction band are separated from the valence band. The band gap amount of energy is needed to free the electrons that can be supplied by particles of light, called photons. As sunlight hits the solar cell, photons from the sunlight hit the valence electrons causing the bonds to break and move the electrons to the conduction band. When electrons move, they are led through an external circuit which is in contact with the conduction band. Running the external circuit makes the electrons lose their energy, and they are returned to the solar cells valence band via another contact. As the electrons return to the valence band, they have again the same energy as in the beginning. The process in which the electrons move through this external circuit is called electric current. The potential that the electrons make to the external circuit is a little less than the band gab energy that was

Table 2.1. *The major advantages and disadvantages of photovoltaics [1]*

Advantages of photovoltaics	**Disadvantages of photovoltaics**
Fuel source is vast and essentially infinite	Fuel source is diffuse (sunlight is a relatively low-density energy)
No emissions, no combustion or radioactive fuel for disposal (does not contribute perceptibly to global climate change or pollution)	
Low operating costs (no fuel)	High installation costs
No moving parts (does not wear)	
Ambient temperature operation (no high temperature corrosion or safety issues)	
High reliability in modules (> 20 years)	Poor reliability of auxiliary (balance of system) elements including storage
Modular (small or large increments)	
Quick installation	
Can be integrated into new or existing building structures	
Can be installed at nearly any point-of-use	Lack of widespread commercially available system integration and installation so far
Daily output peak may match local demand	Lack of economically efficient energy storage
High public acceptance	
Excellent safety record	

needed to make the electrons excitation. Produced power is the voltage times the current created by the moving electrons on the external circuit.

Sunlight creates photons with different amounts of energy. This range of different photons is called a spectrum. The photons which have greater energy than the solar cell's band gab energy are able to push electrons from the valence band to the conductive band and therefore create electrical power. When solar cells are hit by photons which have less energy than the band gab, the energy is transformed into heat. Because of this, the solar cells are generally warmer than the ambient temperature, usually about 20 to 30 °C. The advantage of photovoltaic solar cells is that they do not need high temperature or moving parts to operate and generate electricity.

Understanding and modeling solar cells is easier when thinking of the pn junction in the cell. The pn junction is made by doping so that one side becomes positive and the other negative. By positive and negative sides, it is meant that the side in question has a lot of that type of charge. These sides are the conduction band and valence band contacts in the solar cell.

One semiconductor material is silicon which is one of the most abundant materials in the Earth's crust. Today it is still mostly used in crystalline form in photovoltaic applications. There are other, more efficient materials to absorb sunlight spectrum, but most of them are still under development and research. Solar cells may operate under concentrated sunlight, which means that in order to reduce solar cell expenses, it is possible to gather the sunlight with lenses or mirrors from a greater area, but this would complicate the system. With concentrated sunlight technology there are specially designed materials, but those are still in the demonstration stage.

In practical applications, solar cells are not used alone; instead, they are connected together. The number of solar cells connected together is called a photovoltaic module. The reason for connecting multiple cells together is to increase the total power output and achieve a module powerful enough for practical uses. Photovoltaic modules create electrical DC current which can be combined with rechargeable batteries or inverter to create AC current.

2.2. Photovoltaic power plants

Electricity from Sunlight: An Introduction to Photovoltaics by P.A Lynn [2] supported the information in the following section.

2.2.1. Solar radiation

Power from the Sun to the Earth is approximately 10^{17} W which equals approximately the power of one million modern fossil fuel or nuclear power stations. However, in reality most of the solar radiation hits the oceans and some is blocked by clouds. Solar constant which is used to describe the Sun's power density just above the Earth's atmosphere is 1366 $\frac{W}{m^2}$. Yet on Earth's surface at sea level on a clear day it is no more than about 1000 $\frac{W}{m^2}$ because of about 30 % reduction caused by Earth's atmosphere. This value is widely used as a standard for testing and calibrating photovoltaic cells and systems on Earth.

The annual mean insolation is defined as the average power density received during the whole year. The Earth is spherical with a total surface area of $4\Pi R^2$ (where R is radius) which equals that the annual insolation just before Earth's atmosphere is 342 $\frac{W}{m^2}$. It is, however, divided unequally and a lot less at polars on the earth because of the different angle of sunlight. The average insulation on the Earth's surface differs locally because of atmospheric differences. Because of these conditions, the real annual mean insolation ranges from around 300 $\frac{W}{m^2}$ in the Sahara Desert to less than 80 $\frac{W}{m^2}$ near the poles. From the annual mean insolation we can calculate the total energy received in a particular location over the whole year. This can be done by using the number of hours per year (8760) and multiplying it with the annual mean insolation of the particular location. This information is useful when designing photovoltaic systems. However, it must be noted that these values can change each year.

The Sun's spectral distribution is the range and intensity of the wavelengths in its emitted radiation. This is a considerable matter when considering of different solar cells, because different types of materials respond to different wavelengths in sunlight. The Sun's spectrum resembles black body spectrum when its temperature is about 5250 °C. [3] The following Figure 2.1. demonstrates different types of light wavelengths. Marked in yellow is the spectrum created by the sunlight before it passes the Earth's atmosphere (also known as AM0, which stands for Air Mass 0) and near to that is the 5250 °C black body spectrum which is marked as a black curve. The most important wavelength is the one that arrives to the Earth's surface (also known as AM1.5 which represents the sun 48 ° overhead) marked in red in the figure. Earth's surface spectrum has deep slopes at certain wavelengths because of the absorption caused by oxygen, water vapor and carbon dioxide. What is noteworthy here is how much of the whole spectrum lies within ultraviolet, infrared and visible to human eye, which is also illustrated in Figure 2.1.

When designing solar cell systems, the light spectrum is not the only factor that needs to be considered on the surface of the Earth. Another factor arises from the fact that a part of the sunlight enters the cell in three components, direct, diffuse and albedo. The direct component is the light that comes from the Sun directly into the cell. The diffuse component is made of the light scattered by clouds and dust particles in the atmosphere. The albedo component is made of the light that is reflected from the ground, trees, buildings and other land objects. On a cloudy day and when the panel is not pointed directly to the Sun, diffuse components can be dominating. Albedo components can be dominating in certain places like the Swiss Alps because the fallen snow is reflecting sunlight. The different sunlight components are shown in Figure 2.2.

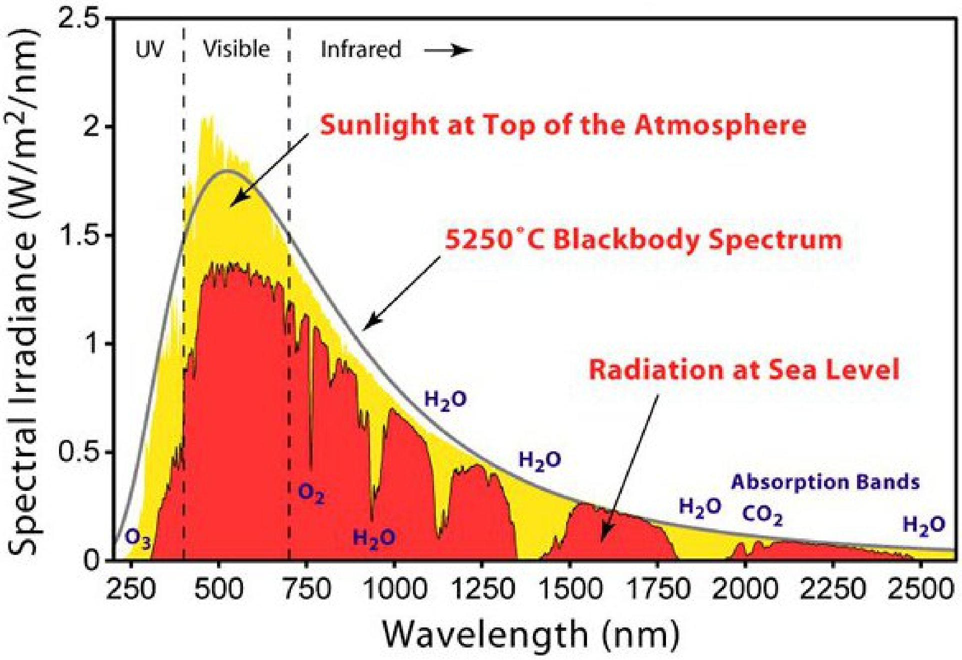

Figure 2.1. *Solar Radiation Spectrum [3]*

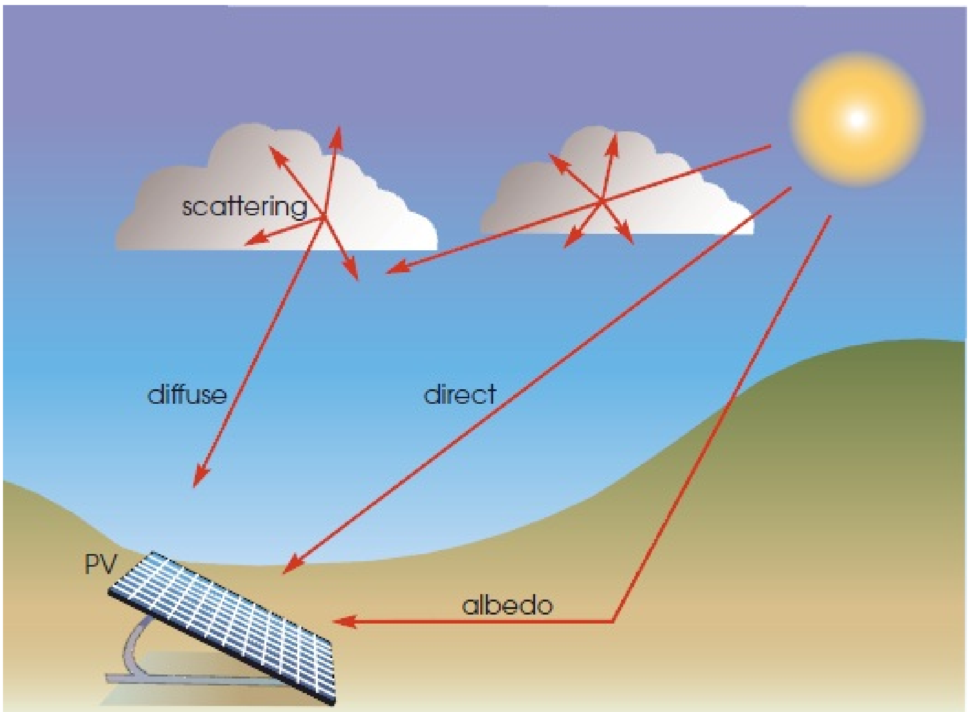

Figure 2.2. *Direct, diffuse and albedo sunlight components [2]*

2.2.2. Photovoltaic modules and arrays

Solar cells are normally never used alone. One cell is typically a low-voltage and high-current device that creates around 0.5 V open-circuit voltage. An acceptable solar module to recharge 12 V battery would usually consist of 36 crystalline silicon cells in series. This kind of module would have around 20 V open-circuit voltage and 17 V maximum power point voltage and would therefore give a good margin to charge a battery even in weak sunlight. Today's trend with high-power grid-connected systems is to create higher output voltages in modules by increasing the number of cells in one module. Because the cells are connected in series, the module output power is limited by the cell that has the lowest output power. The lowest output power is usually with the cell that has the greatest manufacturing fault or if the cell is shaded.

An array of modules is a large number of modules connected together. Single solar cells are usually always connected in series to achieve the greatest possible voltage output, but modules can be connected in series, parallel or as a mixture of both. Adding modules in series will multiple the array voltage output whereas adding them parallel will multiple the array current output. The output power is reduced in the same way as it is with solar cells if added in series. In addition if different modules from different producers are connected together, there would be more losses because of the different I-V characteristics and spectral response times between the modules. Typically on a domestic roof solar array would consist of 10 to 20 modules.

The module efficiency is usually measured in standard conditions in bright sunlight where irradiance is 1000 $\frac{W}{m^2}$, temperature 25 °C and spectrum is AM1.5. Typical values for the most used modules on the Earth are 12 to 16 % for monocrystalline silicon, 11 to 16 % for multicrystaline silicon, 8 to 11 % for CIS and CdTe, and 6 to 8 % for a-Si. This effeciency does not include the module performance when the diffuse or albedo irradiance is in a major role or when the temperature differs from the standard measuring temperature.

2.2.3. Photovoltaic array aligning

The angle how a photovoltaic array is aligned to the sun affects the effectiveness of the array and therefore cost-effectiveness. That is why it is important to know how the Sun is located towards a certain place at certain times and how its path varies. The Earth's path around the Sun is a slightly elliptical and as a result, the distance between the Earth and Sun varies. The distance from the Equator also matters: the further the location is from the Equator the lower the Sun's path through the sky.

Shadows created by building or any other land objects play a critical role in photovoltaic array aligning. If shadows are casted on the photovoltaic module, the efficiency drops dramatically, and it might even damage the system. Occasional shadows like bird droppings, dust or snow are not so critical and can be taken care by scheduled maintenance. However, recurring shadows that are caused by local features are far worse. Setting photovoltaic modules has to be carefully planned because these recurring shadows differ during the year due to the Earth' circulation around the Sun.

When aligning a fixed photovoltaic array, the Earth's circulation should be taken into consideration. With this information the optimum alignment for the arrays to achieve the best possible efficiency can be simulated. The direction of the array should be to the south towards the midday Sun if possible, and if not because of land objects, it should not deviate from south more than 30 °. The horizontal angle should be so that the Sun would make a normal to the panel at noon. Because of the changes in the Sun's noon elevation during the year, it should be decided when the array meets this condition. Usually two equinoxes (about March 21 and September 21) are used, and on these dates the array points straight into the Sun at midday. In Tampere the optimum angle towards the sky for photovoltaic modules is towards south around 45 ° to the horizon. Aligning becomes more complex when diffuse and albedo components of irradiance are taken into account.

2.2.4. Grid-connected photovoltaic systems

Grid-connected (also known as grid-tied) systems are interfaced to an electricity grid. After the 1990s photovoltaic power plants and installations connected to an electricity grid have started to take a more and more dominant role in the market. Since the year 2000 stand-alone systems were overtaken by grid-connected systems in global market, and by 2009 more than 95 % of the production of the solar cells was sold to grid-connected systems. The advantage of a grid-connected system is that when the electricity produced by the photovoltaic array is not needed locally, it can be pushed into the grid. When there is a locally creater need than the array can produce, this extra amount can be imported from the grid.

Inverters are used for converting the DC electricity produced by the solar array to AC electricity that can be pushed into the grid. Inverters are power electronic devices that are able to make AC with correct frequency and voltage to match the grid. Because of the different sunlight conditions caused by, for example, clouds, the inverters must be able to handle a wide range of outputs of the photovoltaic array system, which increases the complexity of the inverters. Usually inverters optimize the energy yield by using the maximum power point tracking (MPPT). Conversion efficiency from DC to AC up to 98 % can be achieved with inverters except when the inverter is operating below 25 % of its maximum power rating. This can be exploited by altering the number of inverters operating at certain times, which means, for example, fewer inverters in use during night time.

A complete grid-connected photovoltaic system consists of several different items other than solar cells and inverters. Depending on the system and the size, up to 50

% of the costs can result from parts other than the solar cells. Module and array structures are required for solar modules, and often they must be customized so that they fit into the target enviroment. The target enviroment can vary from ground to different shapes of roofs. Usually structures are made of aluminium, stainless or galvanised steel. Cables are required to connect the modules to the inverters. In general, the cables are usually double-insulated with UV and water protection. The cables are often sized so that the voltage drop in the cables stays under 2 %. As for a junction box for different module strings there needs to be a photovoltaic combiner unit. Often strings are in parallel and there is a fuse for each string. If a photovoltaic combiner unit does not include DC isolator, there must be a protection unit. With a protection unit it is possible to disconnect the photovoltaic system from the inverter for the times of maintenance and testing. In addition, this protects the system against the damage caused by lightnings. The photovoltaic combiner should be at an easily accesible location. The last application in the list is the energy-flow metering device. This device is used to record the flow of electricity from and to the grid. Typical grid connection core items are shown in Figure 2.3.

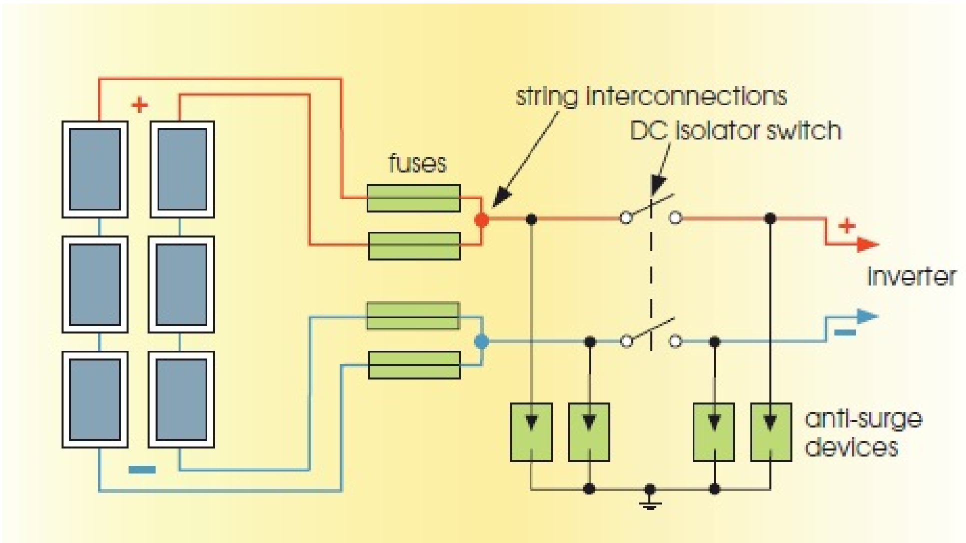

Figure 2.3. *Grid-connected system core items [2]*

2.3. Distributed systems

In the book Distributed Systems Principles and Paradigms by Andrew S. Tanenbaum and Maarten Van Steen there is a good definition of a distributed system: “A distributed system is a collection of independent computers that appears to its user as a single coherent system.” [4, p. 2.] In other words, this definition tells us that the distributed system shows itself to the user as a single system while it actually is multiple computers linked and cooperating together.

The main developing idea in distributed systems is how the different computers work together and how they are connected. As the computers appear to user as one, the communication between them is mostly hidden from the user. These computers don't have to be the same kind, meaning that computer specifications and, operation systems may wary and so on.

Distributed systems should be rather simple to expand in practice. In distributed systems when parts are replaced or fixed or when new ones are added, users and applications should not notice. The information in this section is modified after Distributed Systems Principles and Paradigms by Andrew S. Tanenbaum and Maarten Van Steen. [4]

2.3.1. Goals

It is not always a good decision to construct a distributed system when it is possible. This sub section contains four important goals that should be achieved in order to benefit from a distributed system.

The first and main goal of a distributed systems is about resource access and sharing with users and applications. It should be easy, controlled and efficient. Resources usually include for example printers, computers, storage facilities, data, files, web pages and networks. Resource sharing is economical and often leads to saves. Sharing a printer with everyone instead of buying one for everyone is just one example how to profit from resource sharing. Connecting users and resources leads to easier collaboration and information exchange. An excellent example of this is the Internet and its success.

Today there are also risks with sharing and connectivity. These risks include someone getting a hold of something that does not belong to them and using it. This is why security has an important role in sharing and connectivity technology.

The second goal of a distributed system is related to appearing as a single computer. It has to cover itself mostly so that all physical distribution of its processes and resources is not seen. This is called transparency in distributed systems. Major transparency types are listed in table 2.2.

Table 2.2. *Major types of transparency [4]*

Transparency	Description
Access	Hides differences in data representation and how a resource is accessed
Location	Hides where a resource is located
Migration	Hides that a resource may move to another location
Relocation	Hides that a resource may be moved to another location while in use
Replication	Hides that a resource is replicated
Concurrency	Hides that a resource may be shared by several competitive users
Failure	Hides the failure and recovery of a resource

Here are examples to illustrate each transparency type. For example, access transparency hides the fact that distributed system might have several computers with different operating systems. Different operating systems in this case will very likely create a situation with differences in file manipulation and naming conventions. A simple example of location transparency is the URL *http://www.tut.fi* from which the user cannot see the real location of the TUT web server. Some resource from TUT web server might move into another location without the user knowing it, which is called migration transparency. When the same thing happens while the user is using the file that moves is known as relocation transparency. Replication transparency is shown for example in situations where multiple users are examining a file while actually they are all examining different copies of the same file. This is often necessary to increase availability or performance. Concurrency transparency can be explained by a situation where several users are using the same resource. They should be able to do that without knowing that there are others using the exactly same resource or that it is affecting others in anyway. The resource could be for example a database table with which this happens especially often. An example of failure transparency is when the distributed system is encountering failures and recovering without the user being aware of it.

Complete transparency is not often wanted. There are cases where transparency would do more bad than good, because it can affect performance for example. Sometimes it can even be useful and wanted to be able to see the location. Because of all this, distribution transparency is something that needs to be thought of when designing and implementing the system.

The third goal of distributed systems is openness. In an open distributed system there are standard rules for syntax and semantics of different services. An example of this is computer networks where there are strict rules which govern the format, contents and meaning of sent and received messages. Commonly services in distributed systems are specified through interfaces. It is also important that open distributed systems are extensible. Removing, adding and modifying components should not affect the components that stay in place. This is very well known to be hard in practice.

The last goal in distributed systems is scalability. Scalability can be measured by three different types of dimensions. The first one is size, which means adding users and resources. The second one is geographic, meaning that the users and the resources may locate far away from each other. The third is administration, meaning that managing a distributed system is easy even though it spans many independent administrative organizations. As the system scales up, it is expected to create some performance loss. It is often complicated to design scalable distributed system because it requires much more work to solve all the possible upcoming problems.

2.3.2. Architectures

Architectures in distributed systems are about component organization and interaction. Usually when speaking of distribution system architecture, we are talking about complex pieces of software components distributed into multiple machines.

There are a few important architecture styles, including layered, object-based, data-centered and event-based. In layered architecture style there are multiple layers piled where one layer may only call the layer before it. More loose architecture compared to this is object-based, where different objects may call each other via procedure call mechanism. In data-centered architecture different components interact with data space. In event-based architecture component interaction is based on events.

System architectures are about software components placement. It can be centralized, decentralized or hybrid. Only centralized architecture will be examined here.

Centralized architecture is generally seen as client-server model. In this, clients send requests to the server, the server processes the requests and replies the clients. Communication can be done with a connectionless protocol which is efficient but vulnerable to occasional transmission failures. Another option is connection-oriented protocol which is more reliable but suffers from low performance. For example, a reliable TCP/IP protocol is widely used on the Internet.

In client-server model, there is no clear line which shows what belongs to the client and what belongs to the server. As there is often a database included, the general sketch usually includes three levels which are using the layered style architecture. These three levels are the user-interface level, the processing level and the data level. The user-interface level handles what is required for direct interface with the user, for example how everything is displayed to the user. The processing level mainly includes the applications. The data level manages the data.

Usually the client implements the user-interface level which often has programs to interact with applications. The most simplified user-interface contains only characters while a more complicated one has graphics. Often the server implements the data level, and it can be as simple as a file system, but it is often a database. Mostly in business-oriented environments, the data level consists of a relational database.

The core of an Internet search engine can be used as an example here. A basic search engine works so that the user types a search word which is processed and then gives the result which is a list of web page titles. Processing the search word includes generating a database query to get matching titles, ranking those titles with an algorithm and generating the result HTML. This is illustrated with the three level client-server model in Figure 2.4.

Which part belongs to client and which to server depends on the design and purpose of the system. The more tasks the client has, the more efficient a machine is required for the client, and it works the same way with the server. This is also important regarding the possible extensions or updates because of the required changes will therefore be necessary in the client's computer or in the server. Alternative client-server distinctions are shown in Figure 2.5.

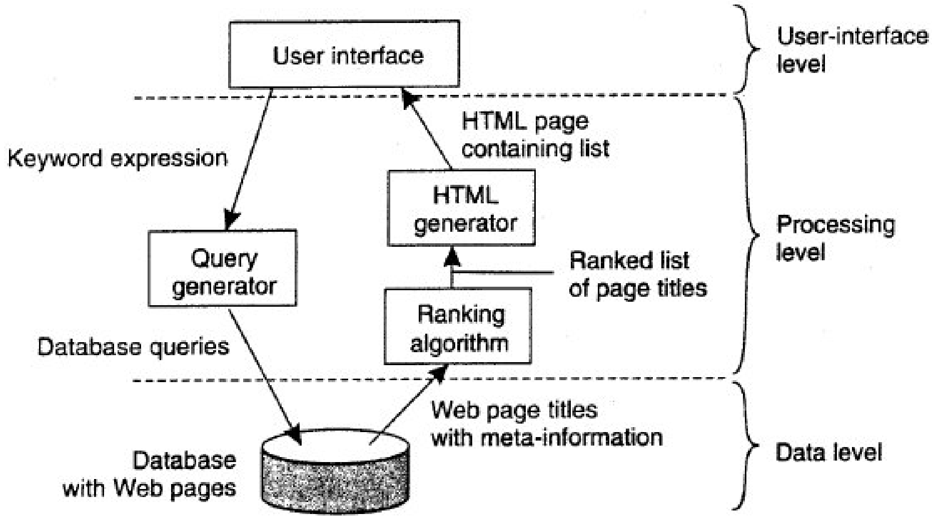

Figure 2.4. *An Internet search engine in three level client-server model [4]*

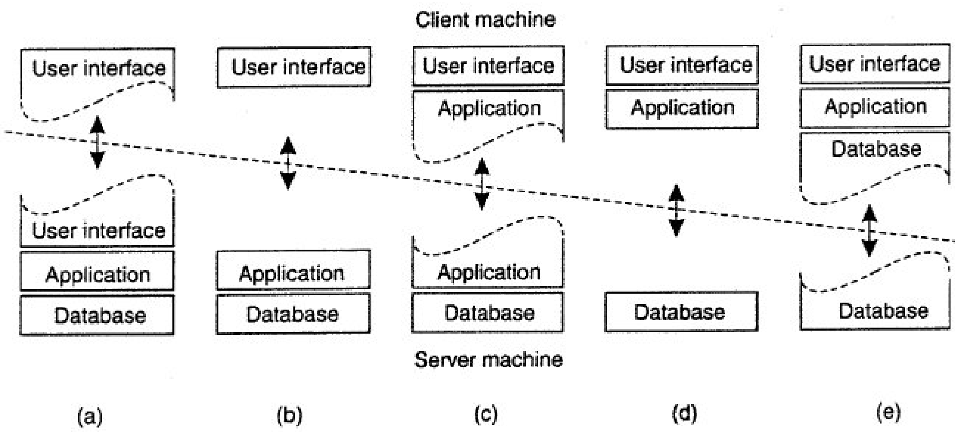

Figure 2.5. *Alternative client-server distinctions (a)-(e) [4]*

2.4. Database systems

Today databases are encountered everywhere even without people noticing them. A simple example is when people go grocery shopping. As they checkout, each item is scanned with a bar code reader. An application program uses this information to fetch the price of the item. After that the program reduces the item count from the database. After the items have been paid for, shop owners can check from the database if some items need restocking.

A database is a collection of related data. Database Management System (DBMS) is the software that manages and controls database activity, and a database application is a program that deals with the database. Database system is a group of application programs, DBMS and the database itself. The information in this section is supported by Database Systems A Practical Approach to Design, Implementation and Management by Thomas Connolly and Carolyn Begg. [5]

2.4.1. Traditional file-based system

A database system is often compared to a file-based system. This is mostly because the file-based system is considered as a predecessor of database system. However, file-based systems are still in use in specific areas where their advantages are superior compared to other systems.

File-based systems are limited in many ways which can be shown with an example. A car shop is a good example environment. In the file-base system each car is filed in different directories at first by their manufacturer, then by year and last by model. Now consider the work that must be done with file-based system when a customer comes and asks for a car which cost less than 5,000 €, has four doors, has less than 200,000 km driven and was build after 2001. In order to make these kinds of searches easier in the file-based system the data must be duplicated some way. This increases the data amount unnecessarily and makes it harder to update because the data need to be updated in several places. File-base system also has data dependence problem. Some data structure changes might require user application changes and existing data files changes. This can be very time-consuming and is highly vulnerable to errors. The last limitation is related to incompatible file formats. Creating data files with one programming language might make them difficult to process with other programming languages. To solve this problem the file needs to be converted into some common format which could be again time-consuming and therefore expensive.

The file-based system works well in certain specific situations. In these situations there are not many items to handle or many items with simple tasks such as store and retrieve. So typically the file system breaks when there is a need to make cross-references or process the information in the files.

2.4.2. Database management system advantages and disadvantages

Database management system (DBMS) has advantages compared to the file-based system but also disadvantages. In this chapter these advantages and disadvantages are explained briefly, beginning with the advantages. The advantages are listed in table 2.3.

The control of data redundancy means that there is no such need of copying the same

Table 2.3. *DBMSs advantages [5]*

Control of data redundancy
Data consistency
More information from the same amount of data
Sharing of data
Improved data integrity
Improved security
Enforcement of standards
Economy of scale
Balance of conflicting requirements
Improved data accessibility and responsiveness
Increased productivity
Improved maintenance through data independence
Increased concurrency
Improved backup and recovery services

data into multiple places as with the file-based system. It is not eliminated completely because there are cases where key items must be in multiple places to model relationships.

Data consistency is an advantage because of controlled data redundancy. Most of the time a data item is stored only once and therefore value updates have to be done only once for the item. If the system holds an item in multiple places and is made aware of it, it automatically handles updates so that each of the places is updated at the same time.

More information from the same amount of data is made available as DBMS can combine different informations. For example, the car shop has general value information of the car and the car manufacturer has more specific information of different parts.

The sharing of data is easier with databases as they can belong to a whole organization compared to files which might be only inside specific computers. New applications can also make use of DBMSs functions and features.

Improved data integrity means that the data are validated and made consistent. The database administrator creates constraints which are consistency rules that DBMS enforces. An example of integrity constraint could be in a car shop where no cars could not be built before 1900.

Improved security means preventing unauthorized users from accessing the database. User authorization is done with user name and password. Database administrator can also define access rights into the database for each user and DBMS enforces these.

The enforcement of standards is basically about defining and enforcing the specific standards, for example, department, organizational, national or international standards for data formats.

The Economy of scale is an advantage created when working with one database as one source of data. This can easily create savings when compared to different file-based systems where each system requires developing and maintaining.

The balance of conflicting requirements is related to different users being in conflict with other users regarding the data. The database can be designed to provide resources for all users in an optimal way.

Improved data accessibility and responsiveness relate to the fact that many DBMSs provide query languages or report writers for users so that they can easily and almost instantly obtain information to their own terminals.

Increased productivity comes from DBMSs implemented functions. In file-based applications the programmer must write these functions, therefore with DBMS the programmer will not have to worry about low-level implementations.

Improved maintenance through data independence is an advantage because in file-based systems application programs are dependent on the data. By being dependent on the data means that data descriptions and logic for accessing the data are built into each application program. Any data changes add massive amount of work to update application program and existing data.

Increased concurrency comes from DBMS as it manages concurrency. This prevents loss the of information and even the loss of integrity.

Improved backup and recovery services are provided in DBMS. These will minimize the amount of data loss in case of failure.

There are not as many disadvantages in DBMS as advantages. Disadvantages are listed in table 2.4.

Table 2.4. *DBMSs disadvantages [5]*

Complexity
Size
Additional hardware costs
Cost of conversion
Performance
Higher impact of a failure

Complexity disadvantage comes from the fact that a good DBMS is an extremely complex software. This means that in order to take full advantage of DBMS, those who work with it must understand it or it can lead to bad design decisions.

Size is a direct response from the complexity. Complex DMBS requires disc space and memory to run efficiently.

Additional hardware cost comes from performance requirements. If a increase in performance is wanted, a hardware upgrade is necessary.

The cost of conversion disadvantage comes from the cost caused by converting existing applications to run on the new DBMS and hardware. This also includes all new manpower and training costs in this operation.

Performance might be better with a file-based system because it is specified for something particular. DBMSs are made to be more general purpose.

The higher impact of a failure comes when the DBMS becomes unavailable. This means that everything tied to DBMS will go down until DBMS is up and running again.

2.4.3. The Relational Model

Relational data model proposed by E. F. Codd (1970) is the base of the relational database management system (RDBMS). [5] RDBMS represents the second generation of DBMSs and is the most dominant data-processing software in use today.

Relational algebra and relational calculus have been defined as the base of relational data model by Codd (1971). [5] Therefore, understanding relational data model requires certain understanding of these two. The relational algebra is a procedural language which is used to instruct the DBMS on how to create new relations from one or more relations in the database. The relational calculus is a non-procedural language which is used to formulate the definition of a relation in terms of one or more database relations.

2.4.4. SQL

Structured Query Language (SQL) has become the standard relational database language. SQL standard was defined by American National Standards Institute (ANSI) in 1986 and adopted in 1987 as an international standard by International Organization for Standardization (ISO). Later the standard has been improved several times with new revisions.

SQL has a few major objects and requirements. First, it allows users to create database and relation structures. Second, users can make basic data management tasks like insertion, modification and deletion of data from the relations. Third, users can also create simple and complex queries. It should not require much of user effort to complete these tasks. A database language command structure and syntax must be easy to learn. Language should follow standard so that it is possible to move to different DBMSs and use the same command structure and syntax.

SQL language is designed to use relations to transform inputs into required outputs. This is also known as a transform-oriented language. SQL does not require access methods to the data be specified. In other words, a user does not need to tell how to get the data, specifying what is required is enough. This also means that SQL is a non-procedural language. SQL is mainly free-format language, meaning that it is not very strict on the placement of parts of statements.

3. TUT PHOTOVOLTAIC RESEARCH POWER PLANT CONFIGURATION AND SPECIFICATIONS

The photovoltaic (PV) research power plant of the Department of Electrical Energy Engineering (DEEE) at the Tampere University of Technology (TUT) consists of a number of different objects. It has several solar panels in different string configurations, various sensors, measuring cards and softwares running all together.

Sensors, panels and hardware choices were researched beforehand and chosen mostly by Diego Torres Lobera. This research was published in his masters thesis. [6]

3.1. Photovoltaic research power plant

The PV research power plant consists of 69 solar panels in different panel strings (Appendix 1.) and a weather station. 21 SP Lite 2 solar sensors and 21 Pt100 temperature sensors have been split between panels in pairs. Each Pt100 temperature sensors have been attached against back of the solar panel. The weather station is split into two physical locations. The first location has a CMP 21 pyranometer with a shadow ring to measure only diffuse irradiance and an HMP 155 sensor to measure temperature and humidity. The second location has a WS 425 wind sensor which measures wind speed and direction and a CMP 22 pyranometer which measures global irradiance. In addition, a thunder rod was built next to the second location because of the wind sensor requirements. Both weather station locations are marked in Appendix 1.

The data from the sensors is first transmitted in the sensor cables and extended from those cables to SF/UTP cables and transferred forward. The SF/UTP cables are twisted pair cables with foil and braiding shields in the cable screening. On the top of the DEEE building, the data cables are protected with an aluminum tube against solar radiation at places visible to the Sun. Without aluminum tubes, the cables would deteriorate significantly faster. Depending on the location, there are from one to three cables inside one aluminum tube.

The sensor data travels from the roof to the data logger inside the DEEE building. Only the weather station data is passed through an air conditional room before the data logger. In the air conditional room, there is a custom designed box in which the data cables are extended once they pass. In the box, there are also DC powers for the second weather station sensors. The first weather station sensors get their DC powers from a custom made box which is under the sensors on top of the roof. The sensor cables are also extended in that box.

Inside the DEEE warehouse room there is a NI 9144 expansion chassis which captures each roof sensor data. SP Lite2 and PT 100 data transmits via direct cables whereas the weather station sensor data comes through the air conditional room into NI 9144. NI 9144 has eight module slots which are occupied by six NI 9217 modules which each take the maximum of four PT100 sensors' data and two NI 9205 voltage modules which take the rest of the sensors' data.

NI 9144 passes the data by RJ-45 Ethernet cable directly to cRIO 9074 in the DEEE lab. cRIO 9074 is connected with an RJ-45 Ethernet cable to the DEEE local Ethernet network. The computers that have access to this local network may communicate with cRIO 9074 with LabVIEW.

One computer was selected as the host computer to communicate with cRIO 9074. This computer fetches the data from cRIO 9074 with LabVIEW, and in the software the data will be pre-processed so that it is ready to be saved into the database. The database has been installed in the same computer so the data is saved in the same computer from the LabVIEW software to the database. The whole system connection scheme can be seen in Figure 3.1.

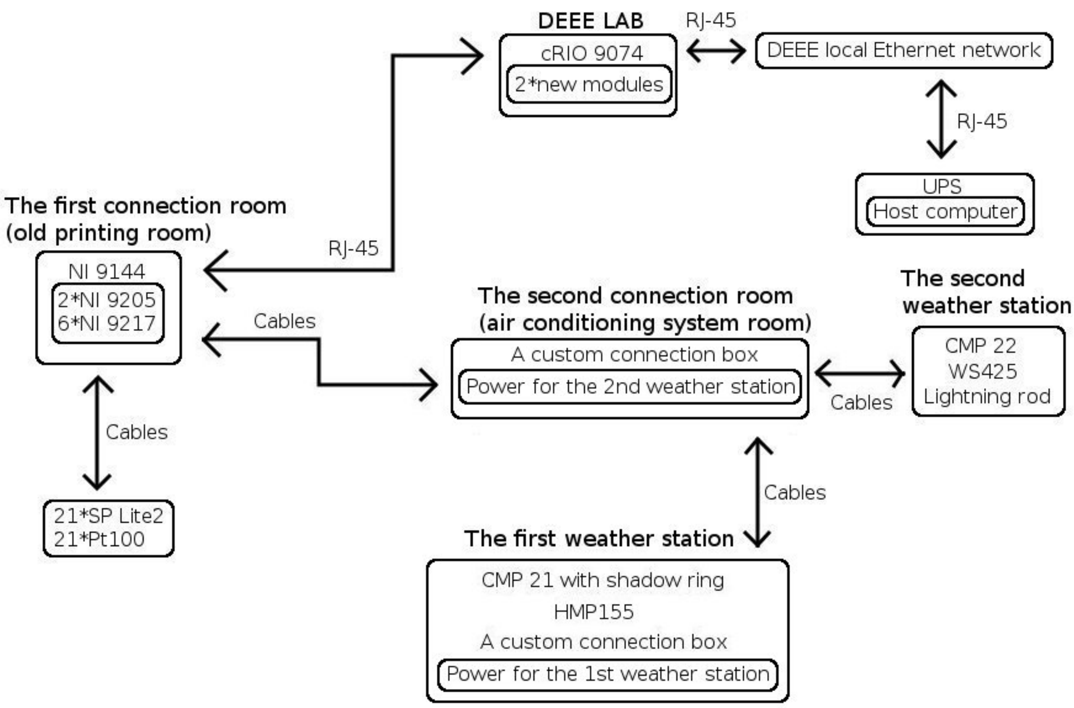

Figure 3.1. *Photovoltaic research power plant measuring and data storing system connection scheme*

3.1.1. Solar panel

TUT DEEE PV research power plant solar panel type is NP190GKg which is manufactured by Naps System Oy. It has 54 polycrystalline Si solar cells in series, and the layout has 6 rows each of which are 9 cells long. Panels are mainly used in grid-connected photovoltaic systems [6]. The picture of NP190GKg solar panel is in the Figure 3.2. The specifications are listed in table 3.1.

Figure 3.2. *Picture of NP190GKg solar panel*

3.1.2. Photovoltaic research power plant layout

The layout designing and construction of the actual PV research power plant solar modules on the top of the DEEE building were finalizes during spring 2010. Modules were designed to be in a grid-connection for research purposes. The main research areas for the PV research power plant are different string connection effects on the generated power, different power electronics in the system, environmental effects on the system and maximum power point tracking.

The PV research power plant consists of 69 NP190GKg panels and the total nominal power of the plant is 12.7 kWp. The panels are in six different strings so that there are three 6 panel strings and three 17 panel strings. The panels were placed so that they would avoid all the possible shadows which might be casted on them at the roof.

The system offers several different outputs for many power configurations. The electrical connections of the strings are illustrated in Appendix 2. The connection scheme for the strings is illustrated in Appendix 3. It has been discussed that two different configurations could be studied in the future. The first is a four-output configuration and the second is a three-output configuration. The first one consists of three strings of 17 solar panels and one string with 18 modules. The second one consists of three strings of 23 modules.

Table 3.1. *Specifications of NP190GKg solar panel [6]*

Module Electrical Performance under Standard Test Conditions (1000 $\frac{W}{m^2}$ solar irradiance, 25 °C cell temperature, Air Mass 1.5)	
Maximum Power Point	190 W, 7.33 A at 25.9 V
Short circuit current	8.02 A
Open circuit voltage	33.1 V
Efficiency ratings	module 13.1 %, laminated area 13.2 % cells alone 14.5 %
General	
Length	1475 mm ±2 mm
Width	986 mm ±2 mm
Thickness at edge	35 mm ±2 mm
Weight	19.5 kg ±0.3 kg
Normal operating cell temperature	46 °C
Cell length	156 mm
Cell width	156 mm
Construction	
Top cover material	low iron tempered glass 4 mm
Rear cover material	PVDF-PET-PVDF
Encapsulant	EVA
Other	3 factory-fitted bypass diodes 1 junction box type S1410-2 2 1 m cables 4 mm^2
Offset caused by 10 W heater	< 1 $\frac{W}{m^2}$ for CMP 11 Pyranometer
Power required	12 VDC, 1.3 A (with 10 W heater)
Weight	1.6 kg

The electrical features of both configurations are listed in the table 3.2. It is also possible to connect all 69 panels together as one string. [6] At the moment, the modules of string 5 are separate, and they are called extra modules. In the department laboratory, there is a control panel for crossed connections.

Table 3.2. *Electrical features of different string configurations [6]*

4-outputs configuration		**3-outputs configuration**	
17-panels string		23-panels string	
P_{MPP}	3.1 kW	P_{MPP}	4.2 kW
U_{MPP}	440 V	U_{MPP}	596 V
U_{OC}	563 V	U_{OC}	563 V
18-panels string			
P_{MPP}	3.3 kW		
U_{MPP}	466 V		
U_{OC}	596 V		

3.2. Sensors

The photovoltaic research power plant consists of a wide variety of different types of sensors. In another thesis ordered by TUT DEEE, a market research was carried out before choosing and ordering any of the sensors [6]. Solar module strings have from one to a few fast irradiance sensors to measure irradiance that is normalized on the panel, and these will notice even the temporal shadows casted on them immediately. Near each of these sensors there is also a thermocouple to measure solar panel backside temperature. The weather station has two pyranometers to measure global irradiance and diffuse irradiance. In addition, the weather station has temperature, humidity and wind sensors.

3.2.1. Solar irradiance

SP Lite2 by Kipp & Zonen is a solar irradiation measuring sensor that is used to estimate the power the DEEE solar modules receive. The sensor is designed to work in all weather conditions and it has a specially shaped diffuser that gives an excellent directional response. In addition, the sensor is mostly self-cleaning. A picture of this sensor is shown in Figure 3.3. Sensor specifications are listed in the table 3.3. [7].

SP Lite2 produces the measures from the entire hemisphere that is 180 ° angle of view. It can be used in different angles towards the sky. The measures are produced in low-level voltages after which they are divided with sensor specific sensitivity in order to get the output in $\frac{W}{m^2}$. Its calibration is valid only for unshaded natural light from the Sun and not for artificial light. Sensors have been calibrated at the manufacturer. Calibration factor value for each SP Lite2 sensor is listed in the table 3.4. where sensor location number means the location of the sensor seen in Appendix 1.

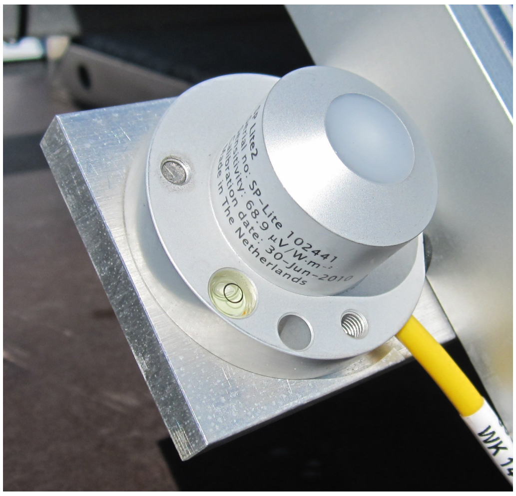

Figure 3.3. *SP Lite2 solar irradiance sensor*

Table 3.3. *SP Lite2 solar irradiance sensor specifications [7]*

Response Time	< 1 second (at 95 %)
Expected signal range	0 to 120 mV
Sensitivity	< ±2 % shift per year
Non-linearity	< ±1 % between 0 to 1000 $\frac{W}{m^2}$
Temperature dependence of sensitivity	< ± $\frac{0.15\%}{°C}$
Spectral range	400 to 1100 nm
Cosine corrected between 80 ° angle of incidence error	within ±5 %
Cosine errors averaged over opposite azimuth error	within ±5 %
Detector type	BPW 34
Weight	110 g
Housing material	Anodized aluminum
Working temperature	-30 to +70 °C

Table 3.4. *SP Lite2 solar irradiance sensor calibration factor values*

Calibration factor ($\frac{\mu V}{W/m^2}$)	**Location**
65.9	1
72.7	2
63.4	3
71.3	4
69.2	5
61.2	6
72.3	7
70.2	8
68.6	9
60.5	10
68.9	11
61.4	12
68.0	13
64.2	14
67.7	15
73.3	16
66.9	17
69.3	18
66.9	19
69.3	20
67.8	21

It was decided when installing SP Lite2 that the sensor should be aligned to the same direction as the panels. This way they measure exactly the same irradiance the panels receive. It is suggested to connect SP Lite2 in differential mode instead of single-ended mode to reduce noise. This was tested in TUT DEEE's PV research power plant, and in single-ended mode there was too much noise in the signal so it was decided that differential mode would be used.

The surface of the SP Lite2 sensor should be cleaned with soft cloth and water, or alcohol, if necessary. It should be checked and possibly re-calibrated every two years. In order to check the calibration, the SP Lite2 should be running for two days with a reference sensor after which the readings should be compared. The reference sensor can be an unused sensor of the same type or a higher grade pyranometer. It is advisable to return the sensor for recalibration if the results differ more than 5 %.

CMP22 and CMP21 by Kipp & Zonen are the irradiance sensors that are used to measure global and diffuse irradiances in this system. The difference between these two is that CMP22 uses very high quality quartz domes for wider spectral range, improved directional response, and reduced thermal offsets. They both exceed the requirements for an ISO Secondary Standard pyranometer and are suitable for reference measurements in extreme climates, from desert to the Antarctic. Pyranometers do not require any power because they generate a low voltage of 0-20 mV in relation to the amount of incoming radiation. The output irradiance can be calculated by dividing the output voltage with the sensitivity value. Kipp & Zonen pyranometers are also recommended to be calibrated every two years, sensors have been calibrated at the manufacturer. CMP22 calibration factor value is 9.52 $\frac{\mu V}{W/m^2}$ and CMP21 calibration factor value is 8.89 $\frac{\mu V}{W/m^2}$. [8] A picture of CMP22 can be seen in Figure 3.4. A picture of CMP21 is in Figure 3.5. Their specifications are listed in the table 3.5.

Figure 3.4. *CMP22 pyranometer measuring global irradiance*

CM121C Shadow Ring from Kipp & Zonen is used with the CMP21 pyranometer to measure diffuse radiation from the sky. Shadow ring has to be adjusted after a few days to ensure that the shadow covers the pyranometer dome completely as the elevation of the

Table 3.5. CMP22 and CMP21 pyranometers specifications [8]

Specification	Unit	CMP 21	CMP 22	Definition
Spectral range	nm	285 to 2800	200 to 3600	50 % response point
Sensitivity	$\frac{\mu V}{\frac{W}{m^2}}$	7 to 14	7 to 14	Signal output for 1 $\frac{W}{m^2}$ irradiance
Impedance	Ω	10 to 100	10 to 100	At instrument housing connector
Response time	s	< 5 < 1.7	< 5 < 1.7	95 % of final value 63 % of final value
Non-linearity	%	< 0.2	< 0.2	From 0 to 1000 $\frac{W}{m^2}$ irradiance
Temperature dependence of sensitivity	%	< 1	< 0.5	Variation in range -20 to +50 °C from value at +20 °C
Tilt error	%	< 0.2	< 0.2	Deviation when facing downwards
Zero offset A	$\frac{W}{m^2}$	< 7	< 3	At 0 to -200 $\frac{W}{m^2}$ of IR net radiation
Zero offset B	$\frac{W}{m^2}$	< 2	< 1	At 5 $\frac{K}{h}$ temperature change rate
Operating temperature	°C	-40 to +80	-40 to +80	Storage temperature is the same
Field of view	°	180	180	Hemispherical
Directional error	$\frac{W}{m^2}$	< 10	< 5	At 80 ° with 1000 $\frac{W}{m^2}$ irradiance
Maximum irradiance	$\frac{W}{m^2}$	4000	4000	Level above which damage may occur
Non-stability	%	< 0.5	< 0.5	Variation in sensitivity per year
Humidity	%RH	0 to 100	0 to 100	Relative Humidity
Uncertainty in daily total	%	< 2	< 1	95 % confidence level
Humidity	%RH	0 to 100	0 to 100	Relative Humidity
Housing		AAb.	AAb.	Anodized Aluminum body
Weight	kg	0.6	0.6	

Figure 3.5. *CMP21 pyranometer with CM121C Shadow Ring measuring diffuse irradiance*

solar course changes each day. The ring will intercept a small part of diffuse radiation from the sky and corrections are necessary to compensate this, U-profile ring simplifies this correction. [9] Shadow Ring is presented in Figure 3.4. and specifications are listed in table 3.6.

Table 3.6. *CM 121 C Shadow Ring for pyranometer specifications [9]*

Material	Anodized Aluminum of seawater proof quality, Stainless steel
Weight including pyranometer CM 11	5.8 kg
Ring $\frac{width}{ring}$ radius ration	0.185
View angle	10.6 °
U-profile shadow ring	defines the accuracy of the view angle constant within ±2 %

The aligning of the shadow ring consists of north-south aligning, tilting of the sliding bars and setting of the shadow ring sliding bars. The first two are one time installations, but the last one should advisably be done every two days, though in most seasons it can be done with longer time intervals. North-south alignment will be done for CM121C with following procedure. It is recommended that the aligning is done at 12:00 True Solar Time, also known as local Apparent Time. At this time, the Sun is shining directly from the south or the north, depending on the location. The shadow ring has to be aligned towards the Sun in north south axis. The ring should be aligned so that the light does not hit the inner side of the ring. The purpose of tilting the sliding bars is to move the sliding bars parallel to the polar axis. The angle between sliding bars and the horizontal should be the same as the geographical latitude of the location where the adequate accuracy is $\frac{1}{4}$ °. The setting of shadow ring sliding bars is the only maintenance that needs to be done periodically. It is done by readjusting the position of the ring which is done by setting the sliding bars. If the Sun is shining, the shadow ring can be adjusted by looking at the shadow it creates over the pyranometer, the outer dome should be completely in shadow, and if it is not shining, the adjustment should be done by using the tables 3.7., 3.8., 3.9. and 3.10. The tables have correction factors and sliding bar settings in northern latitude 60 ° which is closest to latitude of Tampere (north 61 °). It should be noted that each value is valid between the dates around the value and twice a year, meaning that there are months and dates twice in one table. More detailed adjusting instructions can be found in the manual. [9]

Table 3.7. *CM 121 C Shadow Ring correction factors and sliding bar setting in northern latitude 60° part 1 [9]*

Month	JAN			FEB			
Day of month	1	17	26	2	8	15	
Correction factor	1.01	1.01	1.01	1.02	1.02	1.02	1.03
Day of month	12	27	18	10	4	29	
Month	DEC	NOV				OCT	
Solar declination	-24	-22	-20	-18	-16	-14	-12
Sliding bar setting	132	120	108	97	85	74	63

Table 3.8. *CM 121 C Shadow Ring correction factors and sliding bar setting in northern latitude 60° part 2 [9]*

Month	FEB		MAR				
Day of month	21	26	3	9	14	19	
Correction factor	1.03	1.03	1.04	1.04	1.05	1.06	1.06
Day of month	23	17	12	7	1	26	
Month	OCT				SEPT		
Solar declination	-12	-10	-8	-6	-4	-2	0
Sliding bar setting	63	52	42	31	21	10	0

Table 3.9. CM 121 C Shadow Ring correction factors and sliding bar setting in northern latitude 60° part 3 [9]

Month	MAR		APR				
Day of month	24	29	3	8	14	19	
Correction factor	1.06	1.07	1.08	1.08	1.09	1.10	1.10
Day of month	21	16	11	5	31	25	
Month	SEPT				AUG		
Solar declination	0	2	4	6	8	10	12
Sliding bar setting	0	10	21	32	42	52	63

Table 3.10. CM 121 C Shadow Ring correction factors and sliding bar setting in northern latitude 60° part 4 [9]

Month	APR	MAY				JUN	
Day of month	25	2	9	16	26	11	
Correction factor	1.10	1.11	1.12	1.12	1.13	1.14	1.14
Day of month	19	13	6	29	19	3	
Month	AUG			JUL			
Solar declination	12	14	16	18	20	22	24
Sliding bar setting	63	74	85	97	108	120	132

Diffuse irradiance measurements using the CM121C shadow ring require correction. This is because the shadow ring, in addition to the direct irradiance, is also blocking some of its diffuse irradiance. The before mentioned adjusting table also contains correction factors, however, these are only valid when the pyranometer is in the horizontal position. These factors can be calculated with specific formulas. The correction factor C has following relation to the intercepted part S of the downward component of the sky radiation

$$C = \frac{1}{1-S} \tag{3.1.}$$

Next marking the sun's declination D, the latitude of the observation site B and angle between the sun at sunrise (or at sunset) and the sun at true noon in the plane of the ring U_0. With these markings the U_0 can be calculated from following

$$cosU_0 = -tanBtanD \tag{3.2.}$$

Relation between zenith angle Z, declination D, latitude B and time (by the hour angle U) is

$$cosZ = sinBsinD + cosBcosDcosU \tag{3.3.}$$

The ring subtending a solid angle from the center of the ring is V. The intercepted part S of the sky radiation can be calculated with following formula

$$S = \frac{2V cosD(U_0 sinB sinD + sinU_0 cosB cosD)}{\pi} \qquad (3.4.)$$

Correction factors are calculated with formulas 3.1., 3.2. and 3.4. V value is 0.185 rad. Because V varies within 2 % in dependence of the declination D, correction factor has $\pm$0.5 % error. Note, the sliding bar parts rising above the horizon influence is ignored in this calculation. [9]

The CVF3 ventilation unit from Kipp & Zonen is used with pyranometers. It improves the reliability and accuracy of the measurements by reducing dust, raindrops and dew on the dome. The ventilation stabilizes temperature and suppresses thermal offsets. The integrated heater can be used to disperse precipitation and melt frost. It has two modes: 5 W-heating mode and 10 W-heating mode. 5 W-heating mode is used in normal conditions to prevent the formation of dew and frost. 10 W-heating mode is used in extreme climates to melt snow and ice. [10]

CVF3 requires very little maintenance, and it is only necessary to check the removable air inlet filter from time to time and clean or replace it when necessary. The air filter is located at the bottom of the ventilator. It can be changed by removing the the cover by pulling it down with both hands. [10] picture of CVF3 is shown in Figures 3.4. and 3.5. and specifications are listed in table 3.11.

Table 3.11. *CVF 3 Ventilation Unit [10]*

Ventilator fan power	5 W continuously
Heater power (selectable)	5 W and 10 W
Operating temperature	-40 to +70 °C
Air temperature rise caused by CVF 3	< 0.25 K with ventilator fan only < 0.5 K with 5 W heater < 1 K with 10 W heater
Offset caused by 10 W heater	< 1 $\frac{W}{m^2}$ for CMP 11 Pyranometer
Power required	12 VDC, 1.3 A (with 10 W heater)
Weight	1.6 kg

CMF2 Mounting Fixture from Kipp & Zonen is used to mount the CMP22 pyranometer and CVF3 Ventilation Unit combination. It has a 350 mm mounting rod and its diameter is 220 mm. [11] It can be seen in Figure 3.4.

3.2.2. Wind speed and direction

Vaisala WINDCAP® Ultrasonic Wind Sensor WS425 B1B2B measures wind speed and direction. It is a heated and maintenance free sensor. The theoretical mean time between failures is 26 years. It compensates completely the effects caused by temperature, humidity and pressure. In addition, the large transducer heads are insensitive to rain. Stainless steel has been used as the standard sensor material, and the sensor can be mounted upside down. The sensor does not require any special maintenance, though it can be tested and verified if it starts to deteriorate in time due to the environment. [12] Sensor picture is shown in Figure 3.6. Sensor specifications are listed in table 3.12.

Figure 3.6. *WS425 wind sensor*

TUT DEEE's WS425 sensor has been deployed on the roof so that it is at the highest possible place when compared to the nearby objects. Ideally, it should be the highest object within 300 m of horizontal radius. It is advised in the manual [12] that a lightning rod must be installed with the tip at least one meter higher than the wind sensor to protect personnel and the wind sensor. More information about the lighting rod can be found in sub section 3.5.2. The sensor can be aligned towards true north or magnetic north, and at the moment the sensor is aligned towards the magnetic north. When aligning the sensor to its magnetic north, the N-S transducer heads of the sensor should be exactly in line with compass so that the S transducer head is closer to the compass. The picture to demonstrate this is shown in Figure 3.7.

The WS425 sensor operates in analog mode where wind speed and wind direction outputs are voltages. The wind speed is from 0 to 1 VDC where 0 VDC is 0 $\frac{m}{s}$ and 1 VDC is 55.88 $\frac{m}{s}$. It is also possible to get the wind speed as a frequency output where the output range would be from 0 to 625 Hz. The wind direction is from 0 VDC to reference voltage where reference voltage is between 1 to 4 V. In this setup, the reference voltage was chosen to be 4 V to increase accuracy. In wind direction 0 V is 0 $^\circ$ and reference

Table 3.12. WS425 wind sensor specifications [12]

Wind Speed	
Measurement range (analog output)	0 to 56 $\frac{m}{s}$
Measurement range (digital output)	0 to 65 $\frac{m}{s}$
Starting threshold	virtually zero
Delay distance	virtually zero
Resolution	0.1 $\frac{m}{s}$
Accuracy (range 0 to 65 $\frac{m}{s}$)	$\pm$0.135 $\frac{m}{s}$ or 3 % of the reading whichever is greater
Wind Direction	
Measurement range	0 to 360 $^\circ$
Starting threshold	virtually zero
Delay distance	virtually zero
Resolution	1 $^\circ$
Accuracy (wind speed over 1 $\frac{m}{s}$)	$\pm$2 $^\circ$
Outputs	
Analog outputs	
wind speed	
frequency	5 $\frac{Hz}{mph}$
voltage	8.0 $\frac{mV}{mph}$
output impedance	10 kohm
wind direction	
simulated potentiometer	0 to V$_{ref}$
reference voltage	8.0 $\frac{mV}{mph}$
output impedance	10 kohm
Response Characteristics	
Maximum reading rate	1 per second
Sonic measurement time	0.2 s
Signal processing time	0.15 s
Response time	0.35 s
General	
Operating power supply	10 to 15 VDC, 12 mA typical (analog)
and for heated model	36 VDC $\pm$10 %, 0.7 A
Operating temperature	
WS425 non-heated	-40 to +55 °C
WS425 heated	-55 to +55 °C
Material	
body and sensor arms	stainless steel
transducer heads	silicone rubber and PVC
Weight	1.7 kg

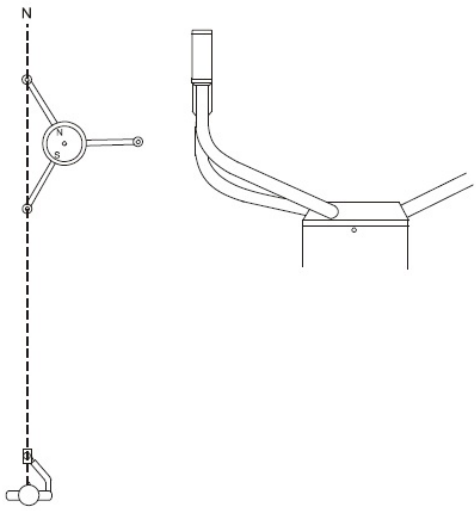

Figure 3.7. WS425 wind sensor aligning to magnetic north [12]

voltage is 360 ° which represents also 0 °. This means that there is a discontinuity point between 359 and 0 ° which is located directly in the north. If the reading is missing, the wind speed output shows the maximum value which in this case is 1 V [12].

3.2.3. Temperature and humidity

Vaisala HUMICAP® Humidity and Temperature Probe HMP155 A4BG12C3A3B1A2A is an instrument with two sensors to measure humidity and temperature. The probe has solid structure and sintered Teflon filter around the sensor to protect against liquid water, dust and dirt. The sensor head is warmed continuously to provide reliable measurements in harsh weather conditions like fog, mist, rain and heavy dew. This makes the humidity level inside the sensor below the ambient level for accurate measurements and reduces the risk of condensation forming on the probe. The probe's fast response time allows measurements in fast changing temperatures. [13] Probes are shown in Figure 3.8. Sensor specifications are listed in table 3.13.

The sensors in TUT DEEE PV research power plant are HUMICAP® 180RC thin film polymer sensor for humidity and resistive platinum sensors (Pt100) for temperature. The sensors are active voltage output sensors with a range from 0 to 10 V. In terms of humidity the voltage is scaled from 0 to 100 % humidity, and with temperature the voltage is scaled from -40 to +60 °C. The probe is a warmed probe with Xheat, and chemical purge is done both periodically and during the start-up. The chemical purge heats the sensor to approximately 180 °C in order to evaporate the interfering chemicals. The whole purge cycle takes about six minutes, and during this time the output values are locked. [13]

The lifetime of HMP155 sensors can be increased by protecting them from scattering and direct solar radiation, and precipitation [13]. In TUT DEEE PV research power plant

Table 3.13. *HMP155 Specifications [13]*

Relative Humidity	
Measurement range	0 to 100 %RH
Accuracy (including non-linearity, hysteresis and repeatability)	
at +15 to +25 °C	±1 %RH (0 to 90 %RH)
	±1.7 %RH (90 to 100 %RH)
at -20 to +40 °C	±(1.0 + 0.008reading) %RH
at -40 to -20 °C	±(1.2 + 0.012reading) %RH
at +40 to +60 °C	±(1.2 + 0.012reading) %RH
at -60 to -40 °C	±(1.4 + 0.032reading) %RH
Factory calibration uncertainty (+20 °C)	±0.6 %RH (0 to 40 %RH)
	±1.0 %RH (40 to 97 %RH)
Response time for HUMICAP 180R(C) at 20 °C in still air with sintered PTFE filter	
63 %	20 s
90 %	60 s
Temperature	
Measurement range	-80 to +60 °C
Accuracy with voltage output	
at -80 to +20 °C	±(0.226 - 0.0028temperature) °C
at +20 to +60 °C	±(0.055 + 0.0057temperature) °C
Operating Environment	
Operating temperature range for humidity measurement	-80 to +60 °C
Storage temperature range	-80 to +60 °C
Electromagnetic compatibility	Complies with EMC std. EN61326-1
Inputs and Outputs	
Voltage output	0 to 10 V
Minimum operating voltage	16 V
Average current consumption	
0 to 10 V	+0.5 mA
During chemical purge	max. 110 mA
With warmed probe	max. 150 mA
Operating voltage	7 to 28 VDC
Settling time at power-up	2 s
General	
Filter	Sintered PTFE
Housing material	PC
Housing classification	IP66
Probe weight	86 g

Figure 3.8. *Humidity and temperature probes of HMP155 [13]*

HMP155 sensors were installed inside DTR503A and DTR502A radiation shields.

DTR503A by Vaisala is a twelve-plate radiation shield which is used with HMP155 humidity sensor. DTR502A by Vaisala is a nine-plate shield used with HMP155's additional thermal sensor. Both radiation shields are maintenance-free and they protect sensors from solar radiation and precipitation. Inside the radiation shield, ventilation is based on the natural air flow created by current wind. The white outer surface of the protection plate reflects radiation and the black surface inside absorbs accumulated heat. [14] Pictures of DTR503A and DTR502A radiation shields are in Figure 3.9.

Figure 3.9. *DTR503A and DTR502A radiation shields with HMP155 sensors inside*

HMP155 sensor can be configurated with an RS-485 serial line or with a USB cable. A USB cable connection does not require that the sensor has power on but requires related software to be installed onto the computer before connecting. [13]

The probe does not require a lot of periodical maintenance, only cleaning the probe with a soft, lint-free cloth with mild detergent, changing the probe filter and changing the sensor when necessary. HMP155 is recommended to be calibrated once a year, and depending on application, it should be checked from time to time. The calibration can be done with a serial line connection or with the buttons on the probe. Before calibrating, the sensor should be chemically purged. It is recommended that the humidity adjustment is done as a two-point adjustment.

The local two-point humidity and temperature adjustment procedure is done in the following way. At first chemical purge should be done after which the protection cover and calibration seals should be removed so that adjustment buttons can be used. After this, the protective plug needs to be opened and three buttons should be seen, see Figure 3.10. Now the button marked with a square symbol should be pressed until a green indicator LED lights up. After that the probe is in RH calibration mode. Next, the filter should be removed and the probe inserted into a dry end reference chamber (for example, LiCl: 11 %RH) in order to make the low humidity offset adjustment. It should be noted that adjustment buttons should not be pressed until the conditions have stabilized which takes approximately 30 minutes. The adjustment can be carried out by using the plus and minus buttons, and after the output voltage is correct, the square button should be pressed until the green indicator LED turns off and back on. The next thing to do is to insert the probe into the high end reference chamber (for example, NaCl: 75 %RH). Again, it is advisable to wait until the conditions have stabilized and then adjust the output voltage with the plus and minus buttons. When the voltage is set, the square button should be pressed until the red indicator LED turns on. Temperature is set next, but setting it can be aborted by pressing the square button twice and the indicator LED should turn off. When setting the temperature, the probe should be in known temperature until stabilized and then again set correct output voltage with plus and minus buttons. Once the output is correct, it can be locked by pressing the square button until a red indicator led turns off and back. The probe is now ready to set the second temperature reference, but it can be aborted by pressing the square button once so that the red indicator led turns off. The second temperature reference of the probe can be set in the same way as the first reference. Once finished, the square button can be pressed and the red indicator led turns off. The HMP 155 manual contains information if one wants to calibrate the sensor in a one-point humidity and temperature adjustment way. In addition, there are details how to calibrate the sensor by using a serial connection. [13]

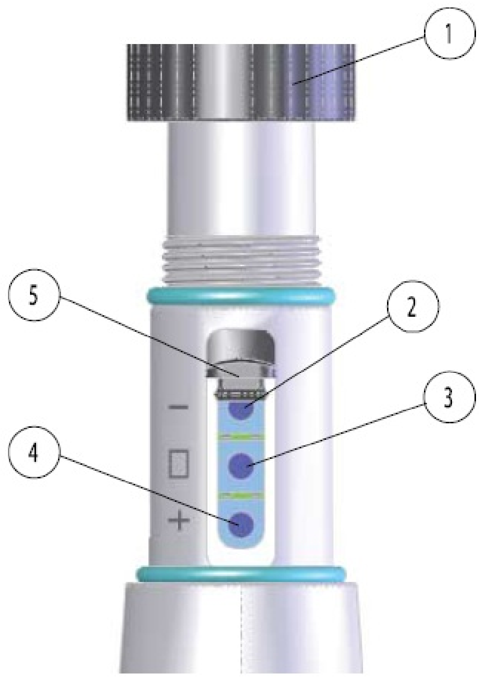

Figure 3.10. *HMP155 from inside [13] where 1 is protective cover, 2 is minus button, 3 is square button, 4 is plus button and 5 is protective plug*

3.3. Data hardware

The main data hardware components are NI cRIO-9074 controller , Ni 9144 chassis, six NI 9217 modules and two NI 9205 modules. There is also a host computer and Back-UPS RS 800.

3.3.1. Data logger

NI cRIO-9074 from National Instrument is used as a data logger in this DEEE photovoltaic research power plant project. It is a real-time controller with reconfigurable field-programmable gate array (FPGA) chassis, and it is for industrial machine control and monitoring applications. NI cRIO-9074 has a 400 MHz real-time processor. FPGA type is Spartan-3 which has 2 million gates. It accepts eight NI C Series I/O modules, including voltage, current, thermocouple, RTD, accelerometer and strain gage inputs. This chassis has dual 10/100 Mbits/s Ethernet ports which allows programmatic communication over the network and connecting expansion chassis. There is also a built-in backup battery to maintain operation for the Real-Time Clock when external power is removed. Ni cRIO-9074 has built-in servers. This means that it can communicate via TCP/IP, UDP, Modbus/TCP, IrDA, and serial protocols. It also offers communication with its built-in servers for Virtual Instrument Software Architecture (VISA), HTTP, and FTP. With the VISA server the user has a remote download and communication access to the reconfigurable I/O (RIO) FPGA over Ethernet. The HTTP server allows the user to use Web browser for HTML pages, files and the user interface of embedded LabVIEW applications. Lastly with the FTP server the user has access to logged data or configuration files. [15] Specifications are listed in table 3.14. A picture of cRIO-9074 is in Figure 3.11.

Figure 3.11. *Picture of NI cRIO-9074 data logger [15]*

NI 9144 from National Instrument is an expansion chassis for the cRIO-9074. It has 8 module slots and FPGA as cRIO-9074. NI 9144 has two Ethernet ports for daisy chaining additional expansion chassis. [16] Specifications are listed in table 3.15. NI 9144 is shown in Figure 3.12.

Figure 3.12. *Picture of NI 9144 expansion chassis [16]*

National Instruments NI 9217 module has four 100 Ω RTD analog inputs to measure temperature. It has 24-bit resolution and built-in 50/60 Hz noise rejection. The module has two different configurable sampling rates. High-resolution sampling rate yields 5 S/s (1.25 S/s per channel) and high-sampling-rate mode yields 400 S/s (100 S/s per channel). NI 9217 is compatible with 3- and 4-wire RTD measurements and automatically detects which one is in use. The module's accuracy error over its whole operating temperature range is less than 1 °C. The module has a channel-to-earth ground double isolation barrier which gives safety, noise immunity and high common-mode voltage range. [17] Six of these modules are attached into NI 9144 slave chassis. Specifications are listed in tables 3.16. and 3.17. Ni 9217 is shown in Figure 3.13.

Table 3.14. *NI cRIO-9074 data logger specifications [15]*

General	
Internal Real-Time Clock Accuracy	200 ppm; 35 ppm at 25 °C
Memory: nonvolatile/system	256/128 MB
Reconfigurable FPGA	
Number of logic cells	46080
Available embedded RAM	720 kbits
Power Requirements	
Recommended power supply	48 W, 24 VDC
Power consumption	20 W maximum
Power supply input range	19 to 30 V
Weight	929 g
Safety voltage (V terminal to C terminal)	35 V max, Measurement Category I
Network	
Network interface	10BaseT and 100BaseTX Ethernet
Compatibility	IEEE 802.3
Communication rates	10 Mbps, 100 Mbps, auto-negotiated
Maximum cabling distance	100 m/segment
RS-232 Serial Port	
Maximum baud rate	115,200 bps
Data bits	5, 6, 7, 8
Stop bits	1, 2
Parity	Odd, Even, Mark, Space
Flow control	RTS/CTS, XON/XOFF, DTR/DSR
SMB Connector	
Output characteristics	
Minimum high-level output voltage / Maximum low-level output voltage	
With 100 μA output current	2.9/0.10 V
With 16 mA output current	2.4/0.40 V
With 24 mA output current	2.3/0.55V V
Driver type	CMOS
Maximum sink/source current	$\pm$24 mA
Maximum 3-state output leakage current	$\pm$5 μA
Input characteristics	
Minimum input voltage	0 V
Minimum low-level input voltage	0.94 V
Maximum high-level input voltage	2.43 V
Maximum input voltage	5.5 V
Typical input capacitance	2.5 pF
Typical resistive strapping	1 kΩ to 3.3 V

Table 3.15. *NI 9144 expansion chassis specifications [16]*

General	
Power Requirements	
Recommended power supply	48 W, 24 VDC
Power consumption	20 W maximum
Power supply input range	9 to 30 V
Weight	906 g
Safety voltage (V terminal to C terminal)	30 V max, Measurement Category I
Network	
Network interface	100BaseTX Ethernet
Compatibility	EtherCAT
Communication rates	100 Mbps
Maximum cabling distance	100 m/segment

Figure 3.13. *Picture of NI 9217 module [17]*

Table 3.16. *NI 9217 module specifications [17]*

General	
Power consumption from chassis	
Active mode	350 mW max
Sleep mode	1 mW max
Thermal dissipation (at 70 °C)	
Active mode	350 mW max
Sleep mode	1 mW max
Weight	142 g
Safety	
Maximum voltage	
All terminals-to-COM	±30 V
Isolation voltages	
Channel-to-channel	None
Channel-to-earth ground	
Continuous	250 V_{rms}, Measurement category II
Withstand	2300 V_{rms}, verified by a 5 s dielectric test
Input characteristics	
Number of channels	4 analog input channels
ADC resolution	24 bits
Type of ADC	Delta-sigma
Sampling mode	Scanned
Measurement range	
Temperature	-200 to 850 °C
Resistance	0 to 400 Ω
Common-mode range	
COM-to-earth ground	±250 V_rms
Channel-to-COM	50 mV
Conversion time	
High-resolution mode	200 ms per channel, 800 ms total for all
High-speed mode	2.5 ms per channel, 10 ms total for all
Noise	
High-resolution mode	0.003 °C
High-speed mode	0.02 °C
Excitation current	1 mA per channel
Noise rejection	
Normal/common-mode rejection	
High-resolution mode	85/170 dB min
High-speed mode	None/155 dB
Input bandwidth (high-resolution)	3.3 Hz

Table 3.17. *NI 9217 module accuracy specifications [17]*

Temperature accuracy (including noise), 4-wire mode (high-speed mode adds 0.1 °C error)		
Measured Value	**Typical (25 °C)**	**Maximum (-40 to 70 °C)**
-200 to 150 °C	0.15 °C	0.35 °C
150 to 850 °C	0.20 °C	1.0 °C
Temperature accuracy (including noise), 3-wire mode (high-speed mode adds 0.1 °C error)		
Measured Value	**Typical (25 °C)**	**Maximum (-40 to 70 °C)**
-200 to 150 °C	0.20 °C	0.50 °C
150 to 850 °C	0.30 °C	1.0 °C

National Instruments NI 9205 module has 32 single-ended or 16 differential analog voltage inputs to measure different voltages. A maximum sampling rate is 250 kS/s and resolution is 16-bit. There are four different input ranges for each channel: ±200 mV, ±1 V, ±5 V and ±10 V. The module has up to 60 V of overvoltage protection between input channels and common (COM) against signal transients. In NI 9205 there is also a channel-to-earth ground double-isolation barrier to give safety, noise immunity and high common-mode voltage range. [18] Two NI 9205 modules are attached into NI 9144 slave chassis. Specifications are listed in tables 3.18. and 3.19. NI 9205 is shown in Figure 3.14.

Figure 3.14. *Picture of NI 9205 module with Spring terminal and D-Sub connectivity [18]*

Table 3.18. *NI 9205 module specifications [18]*

General	
Power consumption from chassis	
Active mode	625 mW max
Sleep mode	15 mW max
Thermal dissipation (at 70 °C)	
Active mode	625 mW max
Sleep mode	15 mW max
Weight	158 g
Safety	
Maximum voltage	
AI, PFI0, and DO-to-COM	±30 V
Isolation voltages	
Channel-to-channel	None
Channel-to-earth ground	
Continuous	250 V_{rms}, Measurement category II
Withstand	2300 V_{rms}, verified by a 5 s dielectric test
Analog input characteristics	
Number of channels	32 single-ended or 16 differential analog input channels, 1 digital input channel, and 1 digital output channel
ADC resolution	16 bits
Conversion time	
R Series expansion chassis	4.5 μs (222 kS/s)
All other chassis	4.00 μs (250 kS/s)
Input coupling	DC
Nominal input ranges	±10 V, ±5 V, ±1 V, ±0.2 V
Minimum overrange (for 10 V range)	4 %
Maximum working voltage for analog inputs (signal + common mode)	Each channel must remain within ±10.4 V of common
Digital characteristics	
Digital input logic levels	
Input high voltage	2.0 to 3.3 V
Input low voltage	0.0 to 0.34 V
Digital output logic levels	
Output high voltage, sourcing 75 μA	2.1 to 3.3 V
Output low voltage, sinking 250 μA	0.0 to 0.4 V

Table 3.19. NI 9205 module accuracy specifications [18]

Accuracy summary				
Nominal Range (V)	±10	±5	±1	±0.2
Absolute Accuracy at Full Scale (μV)	6230	3230	690	174
Random Noise, σ (μV_{rms})	240	116	26	10
Sensitivity (μV)	96.0	46.4	10.4	4.0

3.3.2. Host computer

The dedicated host computer was initially just a work computer but a little more powerful than what they usually are. This is because it was uncertain how efficient a computer this kind of system would require, and it was discussed that once it was known better, the computer could be replaced. The host computer specifications are listed in table 3.20.

Table 3.20. Host computer specifications

Model	Lenovo LE M58p
Processor	Intel Q9500
Memory	4 GB DDR3 1066MHz PC3-8500
Hard drive	320Gb
Operating system	Windows 7 Pro 64bit
Display	Samsung SyncMaster 2443BW

This computer was used to develop and create the whole measuring system for the data from the data logger. So far the computer has not become any kind of bottleneck in the system, and is still being used alone to handle and serve the data.

Because power failures would make the computer to restart, an UPS was ordered later on. The UPS in use is APC Back-UPS RS 800. In addition to the power cables, the network cable also goes trough the UPS to protect the computer. Both the computer and USP are shown in Figure 3.15.

Figure 3.15. *Picture of the host computer and USP*

3.4. Software

Different softwares are a big part of this thesis. This is mostly because the most complicated tasks are related to softwares, and there are numerous tasks completed by using softwares or by making softwares. These different softwares which played a major role in this thesis will be presented next.

3.4.1. LabVIEW

LabVIEW is a system design software. It is used to make measurement and control systems through unprecedented hardware integration. [19] LabVIEW 2011 spring has been used in this system developing and data handling.

LabVIEW has its own unique graphical programming language. This language is called G. Programming G works so that the user wires graphical icons together. After this the graphical code is compiled directly to machine code. Then the code is ready for processors to execute. G language has a lot in common with other programming languages, such as data types, loops, event handling, variables, recursion and object-oriented programming. Graphical code files made with G language in LabVIEW are called Virtual Instruments (VI).

Most text based programming languages work so that there is a sequential series of commands, but G language executes based on the rules of data flow. This is either data-driven or data-dependent. So the execution order is defined by the flow of data between nodes. For example, there are nodes in LabVIEW which consist of three parts: inputs, data processing and outputs. For instance, once the node has valid inputs, the data are processed and outputs are available for the next node if there is such. The nodes may only process the data after their predecessor nodes have finished their own execution.

Graphical programming is considered being easier than text based coding. This is mostly because in graphical programming the programmer can visualize the coding process and there is no need to learn a strict text based syntax. With graphical programming it is also easier and faster to see and visualize the problem and the solution.

LabVIEW's graphical programming has an advantage with debugging because it is easier to follow the debug when the data flow can be seen graphically. LabVIEW also offers always-on compiler which means that the compiler constantly checks if there are errors and offers semantic and syntactic feedback.

There is another advantage with LabVIEW's graphical language compared to sequential languages like C and C++. The code is automatically parallel if the different code blocks can work parallel. The G compiler itself figures parallel sections of codes for execution. Parallelism is important because nowadays processors have more and more cores which means that executing parallel code is faster. With text based languages creating parallel code is more complicated because of the need to implement and manage threads.

With LabVIEW G language one can take advantage of field-programmable gate arrays (FPGAs). FPGAs are silicon chips which can be reprogrammed. They are strongly parallel and have each independent processing task in different sections of the chip. It is important that FPGAs are not limited by the number of processing cores. Programming FPGA with G allows high-speed hardware reliability, customization and tight determinism.

Automatic memory handling is part of G language, meaning that users do not need to allocate memory for anything; G handles it all. LabVIEW offers built-in memory management tools if the user wants to monitor memory usage.

If there is a strong need for text based code for some purpose, it is possible to execute with LabVIEW G language. For example, there is MathScript Node which can be used to include Matlab code into LabVIEW code.

LabVIEW offers one developing environment where users can handle many if not all hardwares. In addition to wide variety of National Instruments' own devices and drivers, they also offer drivers for a third-party hardware.

3.4.2. Database

PostgreSQL is an open source object-relational database system. It works on all main operating systems (Linux, UNIX and Windows). It has native programming interfaces for several different programming languages, including C/C++, Java, .Net, Perl, Python, Ruby, Tcl and ODBC. [20] PostgreSQL has been developed for more than 15 years and during this time it has gained a strong reputation for reliability, data integrity and correctness.

PostgreSQL's SQL implementation is close to the ANSI-SQL:2008 standard, and it has most of the standard's data types. There is also support for storing binary large objects, such as pictures, sounds and video. PostgreSQL fully supports subqueries, read-committed and serializable transaction isolation levels. Its general limits are listed in table 3.21.

Table 3.21. *PostgreSQL database system general limitations [20]*

Limit	Value
Maximum database size	Unlimited
Maximum table size	32 TB
Maximum row size	1.6 TB
Maximum field size	1 GB
Maximum rows per table	Unlimited
Maximum columns per table	250 to 1600 depending on column types
Maximum indexes per table	Unlimited

PostgreSQL is widely in use in Technical University of Tampere. In the past it was generally known that PostgreSQL lacked speed when compared to rival database systems, but this has already changed in the last few major releases. PostgreSQL 9.0 is used as the database system in DEEE PV research power plant.

Liberal license is used in PostgreSQL. This means that it can be used, modified and distributed (open or closed source) by anyone free of charge for any purpose (private, commercial or academic). [20]

3.4.3. MATLAB

MATLAB is a software made by MathWorks company. It is a developing environment for algorithms, data visualization, data analysis and numeric computation. The following information is modified after MATLAB data sheet. [21]

MATLAB has its own high-level language for technical computing. With this language it is possible to solve technical computing problems faster when compared to traditional programming languages, like C, C++ and Fortran. This is because there is no need to do variable declaration, data type specification or memory allocation. MATLAB language has an advantage, because often there is no need for for-loops, which reduces the amount of code to write. Because matrix and vector operations are crucial to scientific and engineering problems, the language has implemented those operations. This also allows fast development and execution. MATLAB is also featured with several traditional programming language features, such as arithmetic operators, flow control, data structures, data types, object-oriented programming and debugging features.

MATLAB allows the user to execute commands without compiling and linking. This will speed up the iteration process when searching the optimal solution. The execution speed of strong matrix and vector computations is improved with processor-optimized libraries. MATLAB has Just-In-Time (JIT) compilation technology for nonspecific scalar computations.

With MATLAB the user can design and develop custom graphical user interfaces. There is an integrated tool for this called GUIDE (Graphical User Interface Development Environment). It is possible to create graphical user interfaces without GUIDE programmatically with MATLAB functions.

There are several toolboxes for MATLAB to expand usability. One related to this thesis is Database Toolbox. With this toolbox the user has functions for data interaction in ODBC/JDBC-compliant databases.

Technical University of Tampere and its Department of Electrical Energy Engineering are using MATLAB widely. It was known that the data gathered from this DEEE PV research power plant system would be mostly processed with MATLAB. The MATLAB version used in this development was 7.12.0.635 (R2011a).

3.4.4. Web server

The Apache HTTP Server is HTTP/1.1 compliant web server. It has plenty of different features and it is a commercial-grade product. The server is highly configurable and can be extended with third-party modules. There is also a customization possibility by writing modules with the Apache module API. Full source code is also provided with the Apache HTTP Server. The server runs with many operation systems, including Windows, Netware 5.x and above, OS/2 and most Unix versions. The Apache HTTP Server is owned by The Apache Software Foundation which is non-profit corporation. [22]

The Apache license allows free download and use of Apache software for personal, company internal or commercial purposes. The use of Apache software in self created packages or distributions is also allowed. More license information is available in official license-faq. [23]

The Apache HTTP Server is widely used, and for example Technical University of Tampere is using it. The Apache HTTP Server 2.2.17 (Win64) was used in web server related developing process in DEEE PV research power plant system.

PHP (PHP: Hypertext Preprocessor) is a popular open source general-purpose scripting language. It is tailored for web development and can be embedded into HTML. Main purpose of the language was to offer web developers a way to write dynamically generated web pages quickly, but with PHP developers are able to do much more. [24]

PHP can be used for serve-side scripting, command line scripting and writing desktop applications. The main focus is in server-side scripting. This means that the code is executed on the server. Executing PHP code generates HTML which is seen by the client. This means that the client cannot see what the actual PHP code was and it remains hidden from client.

PHP works with all main operating systems, such as Linux, many Unix variants, Microsoft Windows, Mac OS X and RISC OS. Today PHP supports most web servers, including Apache, IIS and many others. PHP can be used as a module or as a CGI processor. [24] PHP 5.3.6-Win64 version was used along with Apache HTTP Server in this project for developing process. PHP was loaded with PostgreSQL module in order to work together. Also phpPgAdmin 5.0.2 was used to manage database data.

jQuery is a library in JavaScript which is a dynamic scripting language mostly used in web pages to improve user interfaces and dynamics. jQuery makes several things easier for a web developer, including document traversing, event handling, animating and Ajax interactions. Its goal is to change the way people write JavaScript. jQuery supports CSS 1 to 3 and several browsers, including IE 6.0+, FF 3.6+, Safari 5.0+, Opera and Chrome. [25] jQuery 1.5.1 version was used in this project's web design with MIT license.

jQuery UI is an extension to jQuery and therefore it is needed in order to use the extension. jQuery UI extension gives the programmer the abstraction for low-level interaction and animation, advanced effects and themeable widgets. Users can use the homepage to customize the extension before downloading it. [26] jQuery UI 1.8.13 with customization was used in this project's web design with MIT license. For example, Finnish railroad company VR is using this in their home page.

3.4.5. Revision control

Revision control software is used to save all different versions from code projects. Such program can be later used to compare differences between different versions and revert changes in code if needed. Reverting is handy especially when some changes in code break the code completely and comparing the changes helps to find and fix new errors. Revision control software is essential when multiple people work with the same code. It will help to solve conflicts between different code changes in the same location in code. Generally it should be avoided for multiple people to work with exactly the same code at the same time because of conflicts and extra work generated from conflicts. With revision control software, additional information can and should be inserted with each code change to describe code changes. This information can be read later when inspecting code changes and it helps others to understand the code changes.

TortoiseSVN is a revision control software for Windows. It is free to use and licensed under GNU General Public License. It was used in this project to manage all codes and their different versions. It is recommended to continue to use a revision control software in this project.

3.5. Miscellaneous

In addition to all core items mentioned before, there are also two other important items in the system. First there are the cables that are used in all data transfer, and second, there is the lightning rod for safety reasons.

3.5.1. Cables

There are a few different types of cables used in this system. Firstly there are different sensor cables, but because those are not long enough, each had to be extended with another type of cable. The cable used here is SF/UTP cable which is a twisted pair cable with foil and braiding shield in the cable screening. Secondly there are network cables to transfer data between cRIO-9074, NI 9144 and the host computer. The network cable type is RJ-45.

In some locations on the roof the cables are visible to the sun light. Because the sun would make the cables to deteriorate significantly faster and affect the signal, those cables have been covered with aluminum tubes. Depending on location, there are from one to three cables inside one aluminum tube.

3.5.2. Lightning rod

Vaisala WS425 wind sensor required a lightning rod for sensor and personnel protection. Following the wind sensor manual requirements and according to a thesis related to thunder protection [27], a lightning rod was installed next to the sensor. A picture of the lightning rod and the installment can be seen in Figure 3.16.

Figure 3.16. *Picture of the lightning rod next to the wind sensor.*

The lightning rod has a metallic top supported by a carbon fiber rod. The lightning rod is wired to a grounding electrode which offers great grounding conditions. The wiring was made so that there is distance between the wire and the building so that electricity would not leap to the building wall.

In the thunder protection thesis made by Markku Kumpula [27] there is a graph which shows the relation between thunder protection level, object height and protection angle. The protection angle has been taken from this graph so that there would be first level thunder protection offering best possible protection. It was then calculated how high the thunder rod had to be in order to get the selected protection angle.

4. SYSTEM IMPLEMENTATION

The project status on September 23rd 2010 was following: As mentioned earlier, the sensor, panels and hardware choices were already made. Panels had already been installed and cabling was on its way. The installing of SP Lite 2 and PT100 sensors was also in progress.

First and foremost, the system and its related items and their purpose had to be understood. The tools, such as LabVIEW, would be used to communicate with the data logger and the data, and therefore it was essential to learn to use LabVIEW and its graphical programming language.

A work computer was ordered for the system developing after a brief discussion about its type. A desktop computer was chosen because there was no need of moving the computer anywhere as it would at least at the beginning work as the host computer. A desktop computer is also more efficient and can be upgraded more easily than a laptop. In this kind of system the most critical specs are processor and memory which is why they were taken into account when ordering the desktop computer.

4.1. The connections of the devices

The first physical work to be done was the cabling. The cabling of each sensor was extended with an extension cable. Planting these extension cables was already well underway. Outside at places where sunlight could hit the cables, protective measures were taken and the cables were put inside aluminum tubes.

An old printing room was decided to be the place where all the cables would eventually go so cRIO-9074 and NI 9144 both would be installed there. A custom rack was designed and built specifically for everything that would be installed in the old printing room. The rack included National Instrument devices, grounding connection, protection glass cover and power switch. From now on this room will be called the first connection room.

SP Lite 2 and Pt100 sensors were the first ones to connect into the extension cables. These sensors and weather stations' locations are marked in Appendix 1. Their connection scheme is listed in table 4.1.

The next thing to do was to decide how to connect SP Lite 2 sensors and PT100 sensors to NI 9205 and NI 9217 modules. Connecting PT100 sensors to NI 9217 modules was straightforward because the sensors came with a 3-wire connection so it was used with the modules according to the NI 9217 module instructions. SP Lite 2 sensors and NI 9205

Table 4.1. SP Lite 2 and Pt100 sensor connections to the extension cables

SP Lite 2	Extension cable	Meaning
Red	Blue white	Radiation Signal
Blue	Blue	Signal Reference
Shield		Shield
Pt100	**Extension cable**	**Meaning**
Red	Orange	Positive connection side
White	White red	Negative connection side
White	Green	Negative connection side

modules had two choices, using a single ended connection or differential connection. The differential connection would use two connectors for one sensor and would suffer less from noise. The single ended connection would use only one connector for one sensor but would be more affected by noise. The manual recommended using differential connection, but the single ended connection was still tested first to see if it worked. All these connections were finished and ready for testing on December 28th 2010. NI LabVIEW was the tool to test and build the data acquisition. Temperature recording was right, but all SP Lite 2 recordings were showing only noise which is why it was soon decided that differential connection would be tested next. After switching to differential connection, the results were a lot better as now it was possible to recognize the signal from noise. Changing to differential reduced possible voltage measuring devices from 64 to 32 (two NI 9205 modules). So after 21 SP Lite2 sensors 11 were left to be used for other sensors.

During the spring 2011 the weather stations were built. The first one completed were the one with HMP 155 humidity and temperature sensors and CMP21 with CM121C shadow ring which measures diffuse radiation. It was built on the roof of DEEE, not on the highest part nor the lowest part of the roof. For this weather station, a custom closet with a door was made as a connection box for initial connections, status led lights, power supplies for the sensors and power switch. The second weather station was built after the first one later on in the spring. It consisted of a CMP22 pyranometer and a WS425 wind sensor. The location was one of the highest places on the DEEE roof.

The second connection room was built in an air condition system room. All the wires from both weather stations led there and those which hold sensor data were extended to the first connection room. A custom connection box was built for cable connections, power supplies for the sensors of the second weather station, status led lights and power switch.

A WS425 wind sensor had special connections that required extra work. The initial plan when it was ordered was to convert the analog signal with Nokeval AC/DC converter to digital signal. This was because of the misunderstanding with the wind sensors voltage reference requirement for wind direction. Once this was noted and a few calls were made to the sensors provider Vaisala, an extra voltage converter was decided to be made in the DEEE lab. This voltage converter was adjusted so that it converted 12 VDC to 4 VDC. The maximum voltage allowed by the manual for wind reference was 4 VDC, therefore it was set to 4 VDC to gain the maximum precision. It affects precision because using maximum voltage reference gives the widest voltage range for the angle between 0 degree and 359 degree. A picture of this connection box is shown in Figure 4.1. The connection box scheme is shown in Figure 4.2.

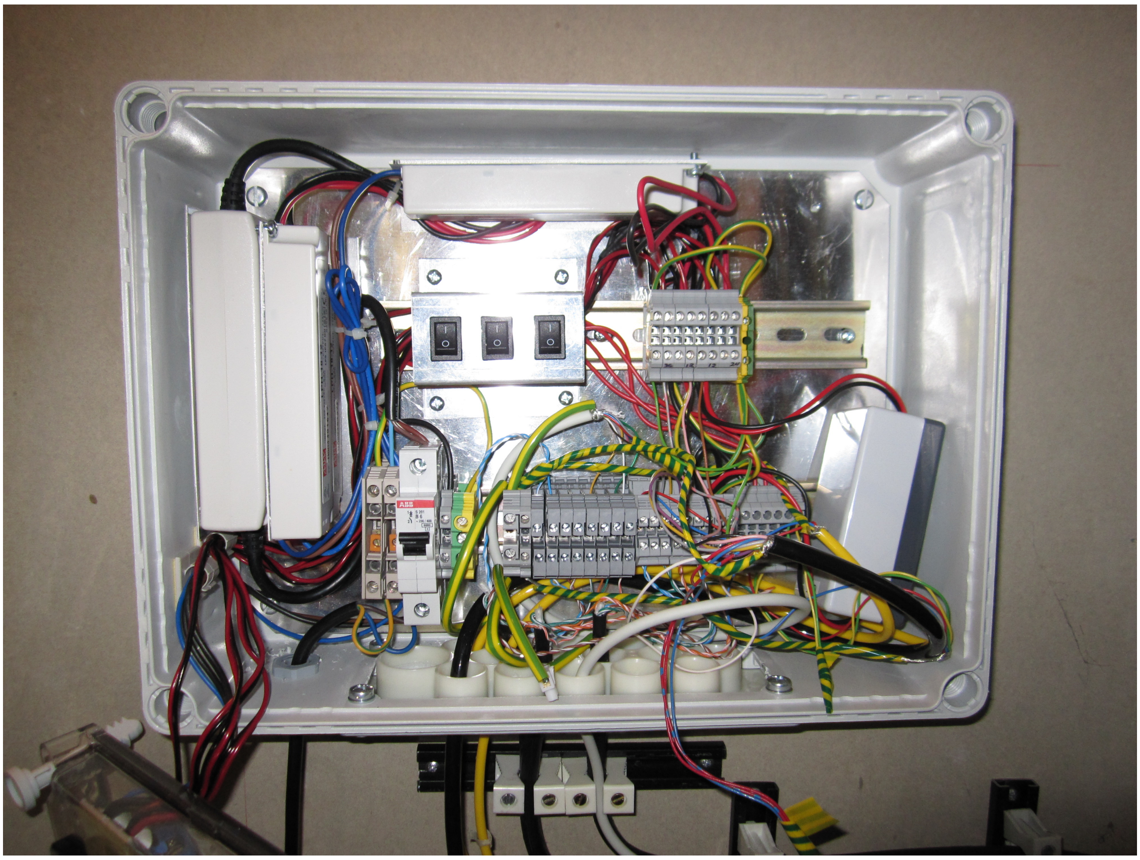

Figure 4.1. *The second connection box open in the second connection room*

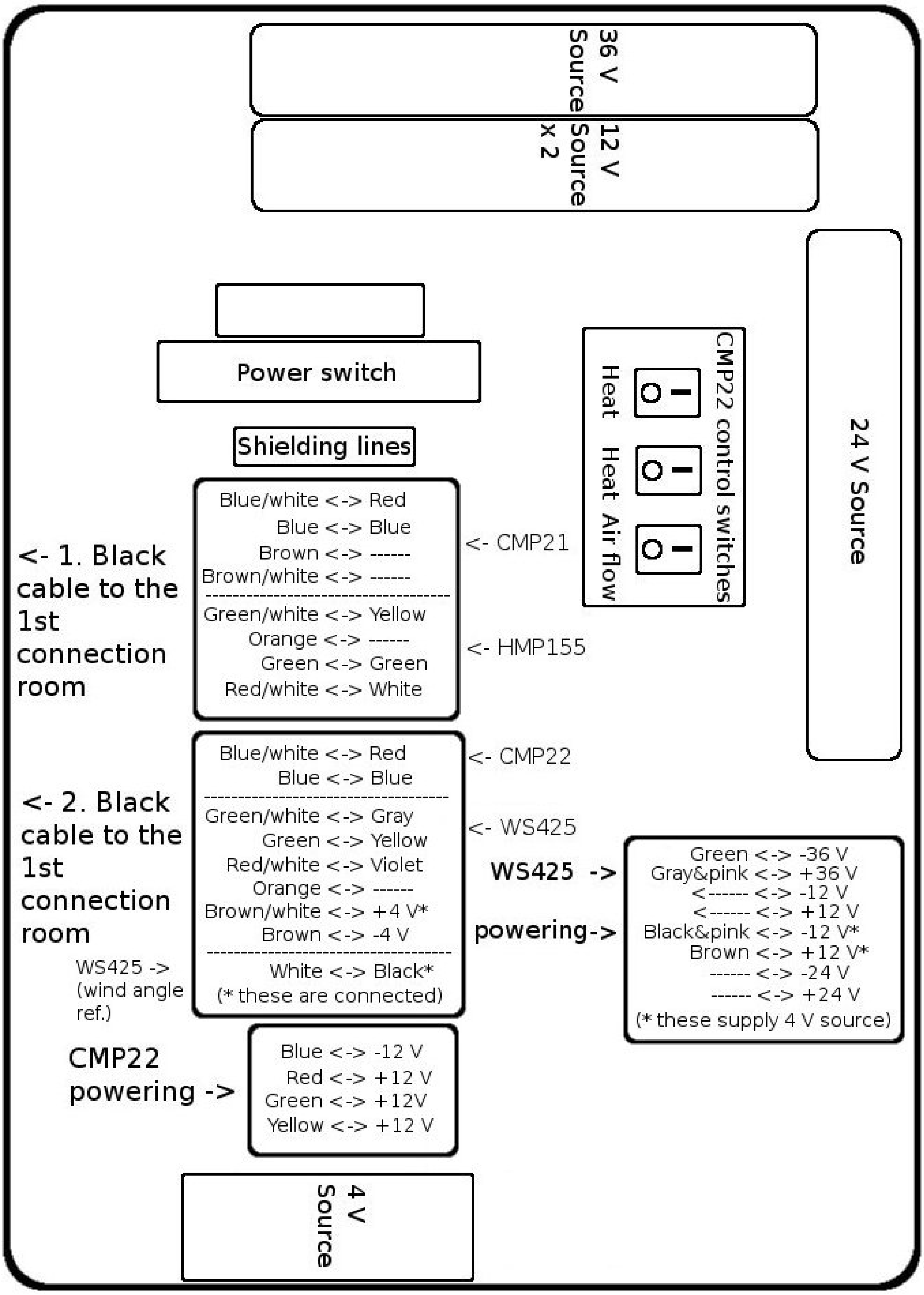

Figure 4.2. *The second connection box scheme*

At the end of the spring 2011, an additional high precision ultra fast voltage module and a digital input output modules were ordered. This caused the following change: cRIO 9074 along with these new modules were moved to the DEEE lab and located inside a new custom rack with a power source and a power switch. After this only NI 9144 along with its power source was left inside the original rack in the first connection room. This change had to be made because the new lab measurements will be done with the new modules and those modules required cRIO 9074 or sampling sent rate would suffer. This change also made it that all the old modules were moved into NI 9144. cRIO 9074 in the new custom rack is shown in Figure 4.3. The first connection room rack with NI 9144, power source and modules are shown in Figure 4.4. The connection list for NI 9144 modules numbered from top down is shown in table 4.2. The connection lists for all modules are in tables 4.3., 4.4., 4.5., 4.6., 4.7. The meanings of the CMP21, CMP22, HMP155, WS425 and CVF 3 wire colors are listed in table 4.8.

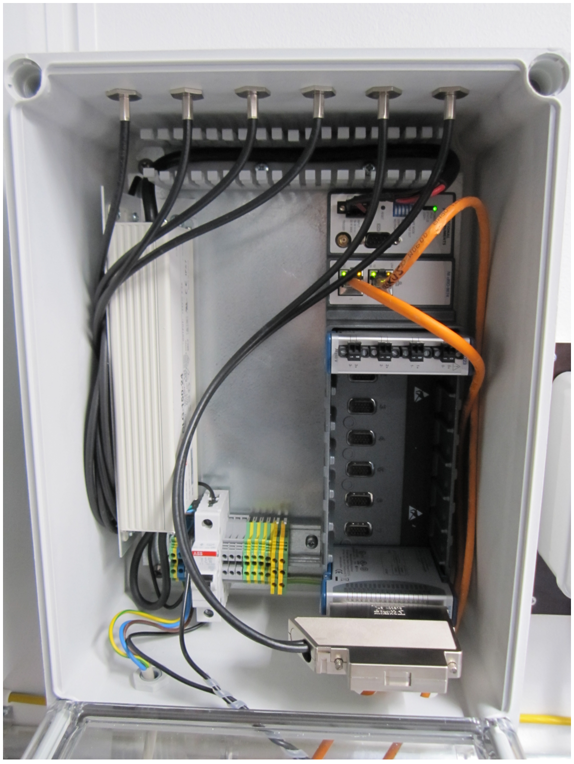

Figure 4.3. *cRIO 9074 inside the new rack in lab*

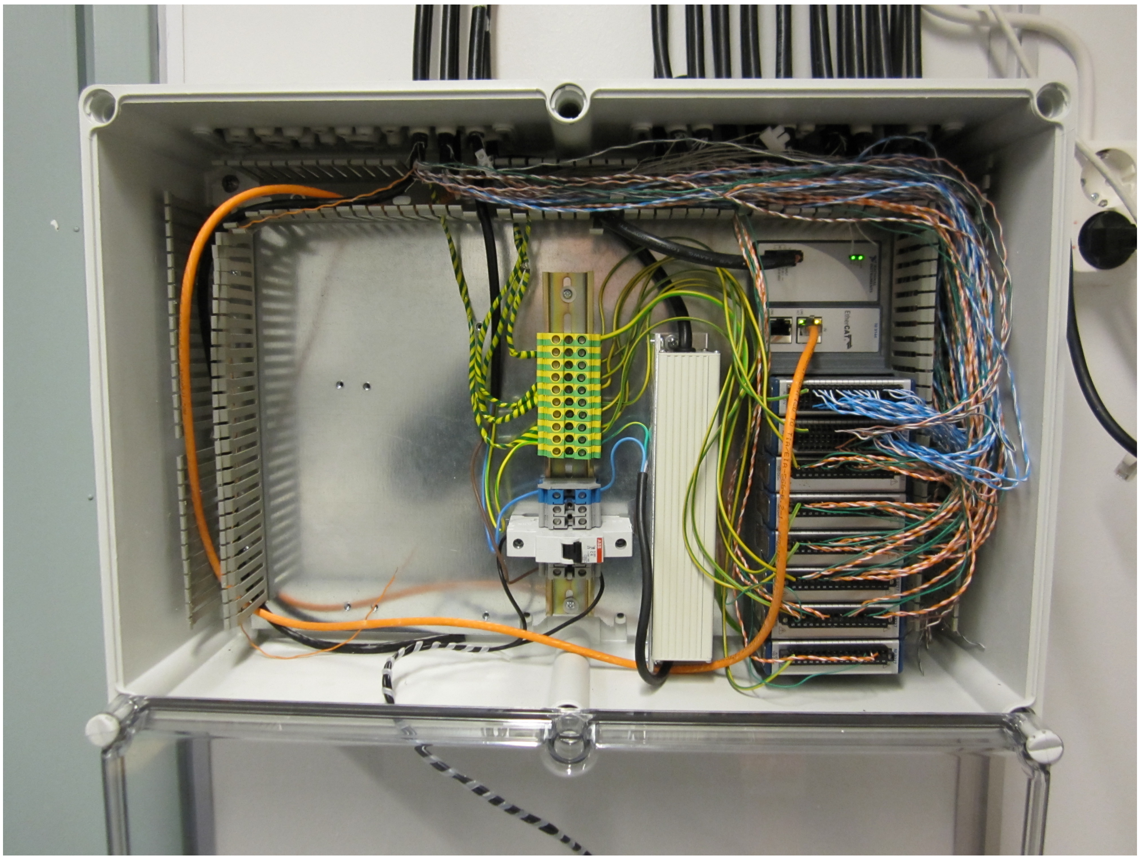

Figure 4.4. *The first connection room rack with NI 9144*

Table 4.2. *NI 9144 modules' connections*

Module	**Type**	**Model**
1.	Voltage	NI 9205
2.	Voltage	NI 9205
3.	Temperature	NI 9217
4.	Temperature	NI 9217
5.	Temperature	NI 9217
6.	Temperature	NI 9217
7.	Temperature	NI 9217
8.	Temperature	NI 9217

Table 4.3. *NI 9144 module 1*

Channel	Cable	Color	Channel	Cable	Color	Sensor
AI0	1	White blue	AI8	1	Blue	SP Lite 2
AI1	2	White blue	AI9	2	Blue	SP Lite 2
AI2	3	White blue	AI10	3	Blue	SP Lite 2
AI3	4	White blue	AI11	4	Blue	SP Lite 2
AI4	5	White blue	AI12	5	Blue	SP Lite 2
AI5	6	White blue	AI13	6	Blue	SP Lite 2
AI6	7	White blue	AI14	7	Blue	SP Lite 2
AI7	8	White blue	AI15	8	Blue	SP Lite 2
AI16	9	White blue	AI24	9	Blue	SP Lite 2
AI17	10	White blue	AI25	10	Blue	SP Lite 2
AI18	11	White blue	AI26	11	Blue	SP Lite 2
AI19	12	White blue	AI27	12	Blue	SP Lite 2
AI20	13	White blue	AI28	13	Blue	SP Lite 2
AI21	14	White blue	AI29	14	Blue	SP Lite 2
AI22	15	White blue	AI30	15	Blue	SP Lite 2
AI23	16	White blue	AI31	16	Blue	SP Lite 2
COM	Ground	Green yellow	AISENSE			
DO0			PF10			

Table 4.4. *NI 9144 module 2 (cables with star(*) are from the second connection room)*

Channel	Cable	Color	Channel	Cable	Color	Sensor
AI0	17	White blue	AI8	17	Blue	SP Lite 2
AI1	18	White blue	AI9	18	Blue	SP Lite 2
AI2	19	White blue	AI10	19	Blue	SP Lite 2
AI3	20	White blue	AI11	20	Blue	SP Lite 2
AI4	21	White blue	AI12	21	Blue	SP Lite 2
AI5	1*	White blue	AI13	1*	Blue	CMP21
AI6	2*	White blue	AI14	2*	Blue	CMP22
AI7	1*	White blue	AI15	1*	Green	HMP155 Hum.
AI16	1*	White blue	AI24	1*	Green	HMP155 Temp.
AI17	2*	White blue	AI25	2*	Green	WS425 Spd.
AI18	2*	White blue	AI26	2*	Green	WS425 Ang.
AI19	2*	White blue	AI27	2*	Brown	WS425 Ref V
AI20		AI28				
AI21		AI29				
AI22		AI30				
AI23		AI31				
COM		AISENSE				
DO0		PF10				

Table 4.5. NI 9144 module 3 and module 4

Module 3.			Module 4.			
Channel	**Cable**	**Color**	**Channel**	**Cable**	**Color**	**Sensor**
EX0			EX0			
RTD0+	1	Orange	RTD0+	5	Orange	Pt100
RTD0-	1	White red	RTD0-	5	White red	Pt100
COM	1	Green	COM	5	Green	Pt100
EX0			EX0			
RTD0+	2	Orange	RTD0+	6	Orange	Pt100
RTD0-	2	White red	RTD0-	6	White red	Pt100
COM	2	Green	COM	6	Green	Pt100
EX0			EX0			
RTD0+	3	Orange	RTD0+	7	Orange	Pt100
RTD0-	3	White red	RTD0-	7	White red	Pt100
COM	3	Green	COM	7	Green	Pt100
EX0			EX0			
RTD0+	4	Orange	RTD0+	8	Orange	Pt100
RTD0-	4	White red	RTD0-	8	White red	Pt100
COM	4&G	Green&G/Y	COM	8&G	Green&G/Y	Pt100

Table 4.6. NI 9144 module 5 and module 6

Module 5.			Module 6.			
Channel	**Cable**	**Color**	**Channel**	**Cable**	**Color**	**Sensor**
EX0			EX0			
RTD0+	9	Orange	RTD0+	13	Orange	Pt100
RTD0-	9	White red	RTD0-	13	White red	Pt100
COM	9	Green	COM	13	Green	Pt100
EX0			EX0			
RTD0+	10	Orange	RTD0+	14	Orange	Pt100
RTD0-	10	White red	RTD0-	14	White red	Pt100
COM	10	Green	COM	14	Green	Pt100
EX0			EX0			
RTD0+	11	Orange	RTD0+	15	Orange	Pt100
RTD0-	11	White red	RTD0-	15	White red	Pt100
COM	11	Green	COM	15	Green	Pt100
EX0			EX0			
RTD0+	12	Orange	RTD0+	16	Orange	Pt100
RTD0-	12	White red	RTD0-	16	White red	Pt100
COM	12&G	Green&G/Y	COM	16&G	Green&G/Y	Pt100

Table 4.7. *NI 9144 module 7 and module 8 (cables with star(*) are from the second connection room)*

Module 7.			**Module 8.**			
Channel	**Cable**	**Color**	**Channel**	**Cable**	**Color**	**Sensor**
EX0			EX0			
RTD0+	17	Orange	RTD0+	21	Orange	Pt100
RTD0-	17	White red	RTD0-	21	White red	Pt100
COM	17	Green	COM	21	Green	Pt100
EX0			EX0	1*	White green	Pt100
RTD0+	18	Orange	RTD0+	1*	Orange	Pt100
RTD0-	18	White red	RTD0-	1*	White red	Pt100
COM	18	Green	COM	1*	Green	Pt100
EX0			EX0			
RTD0+	19	Orange	RTD0+			
RTD0-	19	White red	RTD0-			
COM	11	Green	COM			
EX0			EX0			
RTD0+	20	Orange	RTD0+			
RTD0-	20	White red	RTD0-			
COM	20&G	Green&G/Y	COM			

Table 4.8. *CMP21, CMP22, HMP 155, WS425 and CVF 3 wire color meanings*

CMP22 & CMP21	**Meaning**
Red	Positive
Blue	Negative
Shield	Ground
HMP155	**Meaning**
Blue	Input power voltage
Red	Input ground
Shield	Shield
Brown	RH out
Pink	Analog ground
White	Pt100 positive connection side (inside)
Yellow	Pt100 positive connection side (outside)
Green	Pt100 negative connection side (inside)
Grey	Pt100 negative connection side (outside)
WS425	**Meaning**
Grey pink	+36 VDC
Green	Ground for +36 VDC
Brown	+12 VDC
Black	Ground for +12 VDC
Grey	WD V out
White	WD V ref in
Pink	WS F out
Violet	WS V out
Yellow	WD and WS ground
CVF 3	**Meaning**
Red	+12 VDC ventilator
Blue	12 VDC ventilator ground
Shield	Ground
Green	+12 VDC 5 Watt heater
Yellow	+12 VDC 5 Watt heater

4.2. The grounding of the devices

The system grounding was done based on the principle that each device would be grounded only from one place. This was because if two places were grounded it would form a circuit.

SP Lite 2 sensors were grounded from the structure, and therefore shield lines were cut. All other sensors were grounded from shielding lines and were extended to the first connection room and grounded together there. This means that the shield lines of the weather station sensors were not grounded in the second connection room but continued from there to the first connection room.

The voltage sources in the second connection room were grounded to the connection box structure. Without this, an unexpected AC voltage could have reached the modules and disabled one module.

When it was found out that the wind sensor requires a lightning rod because of personnel and equipment safety, it was decided to build one. There were several discussions with Kari Kannus because of his expertise in this field to build one. The lightning rod was grounded to a grounding electron. This would give the best possible grounding against lightning strikes.

4.3. Data saving in file-based system

File-based systems are partially old technology, but they can still be used in specific cases because of their advantages. A file-based system means that the data are saved into files directly in a hard drive which is usually a network drive if the data are wanted to be shared. The data are usually categorized by different folder names and file names.

The advantages of a file-based system are the small size and fast speed with simple tasks. Both advantages come basically from the fact that in a file-based system the data can be saved into a file in binary form. Binary form is the smallest possible form for any data especially if each data value is of the same size and therefore there is no waste of extra space. This is the case with the present system where all data values will be saved as single-precision floating-point format that occupies 4 bytes (32 bits) (Big-endian order). This format precision is about 7 decimals rounded.

In this system the data would only be saved and later read and analyzed. If it is read and analyzed with an external program it works well, but it does not offer any analyzing features. This was discussed, and it was decided that the data would be mainly analyzed with MATLAB.

Because of these facts, a file-based system was implemented. In this system there is a folder for each year which holds a month folder for each month. Inside the month folder there is a folder for each day, and inside each day there is a file for each sensor data using the earlier mentioned binary single-precision floating-point form. There is also a time file which holds time for each measurement. The time file is in 4 bytes int format (32 bits) (Big-endian order). The time is in milliseconds counted from midnight. In each file the first record is the first after midnight and the last the one before midnight. Here is an example of a MATLAB code snipped how to open one of both types of files in MATLAB.

```
%This example demonstrates how to read binary data files into matlab
fid = fopen('CMP22.bin');
data = fread(fid, inf, 'single', 'ieee-be');
fclose(fid);

%Time is in milliseconds, must be divided to get hours etc.
fid2 = fopen('Time in ms.bin');
time = fread(fid2, inf, 'int32', 'ieee-be');
fclose(fid2);
```

The file creation and data pushing was done in LabVIEW. The solution is detailed in the LabVIEW section later on.

4.4. The database system study and implementation

The database system plays a critical role in this whole system, so it had to be carefully studied and designed based on needs. There are hundreds of different database systems and choosing the one or the ones that would be used in this system demanded studying.

The initial approach was to study free database systems. They do not offer commercial support services which is why the database administrator is burdened more, but they are free. The focus was mainly on relational databases systems but a round-robin database was also studied. The round robin type would be a nice way to handle temporal time based data, for instance a day time period.

There are two widely known open source relational databases systems. The first one is MySQL and the second is PostgreSQL. Sun Microsystems acquired MySQL AB on 26th February 2008 and Oracle acquired Sun Microsystems on 27th January 2010. MySQL Community Edition is still freely available under its GPL license, while other editions have more features and a price starting from 2,000 USD. PostgreSQL is owned by The PostgreSQL Global Development Group. It is available under a liberal Open Source license which is similar to the BSD or MIT licenses.

In the past MySQL was known to be a lot faster than PostgreSQL, but this is no longer the case as nowadays both are equally fast. Choosing one over the other is mostly about personal preference, license, what you are used to use and small feature differences. In Technical University of Tampere PostgreSQL is used for teaching and inside at least some units.

Both database system softwares were tested for this system. PostgreSQL 9.0.4 was used with default settings, and MySQL 5.1.55 had pre-configurated settings to choose, so one matching half of the computer specs was chosen. A half because there would be other softwares like LabVIEW and PostgreSQL running at the computer on the same time. MySQL had different types of engines, and MyISAM, InnoDB and Archive were tested. The archive engine was soon dropped out from the tests because of the lack of indexing which made retrieving data extremely slow as the database size became enormous. Eventually the last tests were run with InnoDB because it was transaction-safe and reduced I/O for common queries based on primary keys.

Both databases were populated with random data in expected form from the system. This was done with LabVIEW software. The data amount was one year's data from all sensors. To gain the maximum performance when populating the databases, multiple thousands of rows were added in one database commit. This process took more than 16 hours but less than 24 hours, depending on the database software. After this operation the PostgreSQL database size was 75 GB and the table size of MySQL database was 69.8 GB, note that MySQL size does not include index. The time type in MySQL database's table was DECIMAL(20,6) (the use of DECIMAL type is explained below) and every other type was FLOAT. The engine in MySQL database table was set as innodb. The time type in PostgreSQL database's table was timestamp and every other type was real. In the tables of both databases each column was declared as NOT NULL, meaning that it has to have a value. The time was also set as the PRIMARY KEY.

Both databases were tested randomly several times with different queries. MySQL had more computer resources in use because of higher settings, which is why it was also slightly faster than PostgreSQL. However, speed was not so high a priority that it would have been needed to perform extensive performance tests with both softwares to find out the slightly faster one. More critical was the fact that MySQL did not support microseconds in timestamp data type. Microseconds were required because of the upcoming extensions to the system which would record data faster than millisecond. However, there was a solution for this MySQL problem, namely to use Time DECIMAL(20,6) data type which requires a total of 13 bytes. Now PostgreSQL offers timestamp format which holds the microseconds and its size is only 8 bytes. Both database systems were tested with MATLAB, too. There was a small naming problem with MySQL because of this DECIMAL(20,6) data type. Because of some unknown reason the name had an hexadecimal value including prefix 0x in it after fetching the time to MATLAB. PostgreSQL did not encounter this issue with its timestamp. Because of these small issues and that PostgreSQL seems more reliable to stay as an open source database system, although it is very unlikely that MySQL would trash community edition, PostgreSQL was chosen to be the database system to be used in this system.

PostgreSQL database backup system for this system was developed at the later stage of this project. PostgreSQL has three different backup ways: SQL Dump, File System Level Backup and Continuous Archiving and Point-In-Time Recovery (PITR). Backup location was ordered from DEEE computer office. It is a network drive location which saves the data incrementally. From PostgreSQL backup methods only the last one supports incremental backuping. If one of the first two backup methods would have been used, the same data would be backed up multiple times which is why the last one was decided to be used.

When the system was completed, the database system load expectations were still unknown. Therefore PostgreSQL settings were left to default. This is something that can be improved in the future when the load is better known and if the database system hardware changes.

4.5. Graphical user interface for data fetching in MATLAB

Since MATLAB would be the main analyzing tool for the data, some way to move the data to MATLAB had to be found. With the file data it was simple since MATLAB commands could be used to get the data from files. The database data, on the other hand, required more planning. One possibility would have been to make a tool which would download data from the database into a file and then MATLAB could extract it easily. The other and the better way was to use MATLAB to make a connection to the database to download the required data.

Programmers can create scrips and graphical user interfaces (GUI) into MATLAB which uses these scripts. Scripts in MATLAB are codes that can be executed in MATLAB. Creating a GUI which would use scripts to connect to the database would be the most user friendly program. MATLAB has GUIDE (GUI development environment) which is not as advanced as common programming language tools such as Java, C++ or C#, but it can do the very basics of GUI creation.

MATLAB has a database toolbox which is an interface to help connecting and performing actions with a database. This was used along with GUIDE to program GUI which would connect to the database and fetch user's selected data into MATLAB for further analyzing. MATLAB also has Visual Query Builder to create SQL statements to fetch data but using it would have been more complicated than custom tailored GUI and it is also more limiting than a database toolbox.

The created GUI makes a connection to the database with the help of a database toolbox. In order to get the database toolbox to work, database JDBC drivers had to be installed and added into MATLAB. Another solution would have been using ODBC connection, but this actually works so in MATLAB that it uses ODBC and then again JDBC drivers to make the connection; therefore using JDBC is a better option. The tested MATLAB version also refused to work in the system with ODBC. MATLAB offers two ways to add JDBC drivers, the first is dynamically loaded at runtime and the second is by adding the driver into MATLAB classpath as a static. If added in a static way, the driver class will load somewhat faster. It should be noted that the static way requires a restart of MATLAB in order to take effect.

The next task for the GUI is to use user selected choices to form SQL language. This language is used and executed with the help of a database toolbox.

The last thing the GUI does is adding the result into MATLAB data variable. After this, the database connection is closed so that it is not left accidentally open. A picture from the GUI is in Figure 4.5.

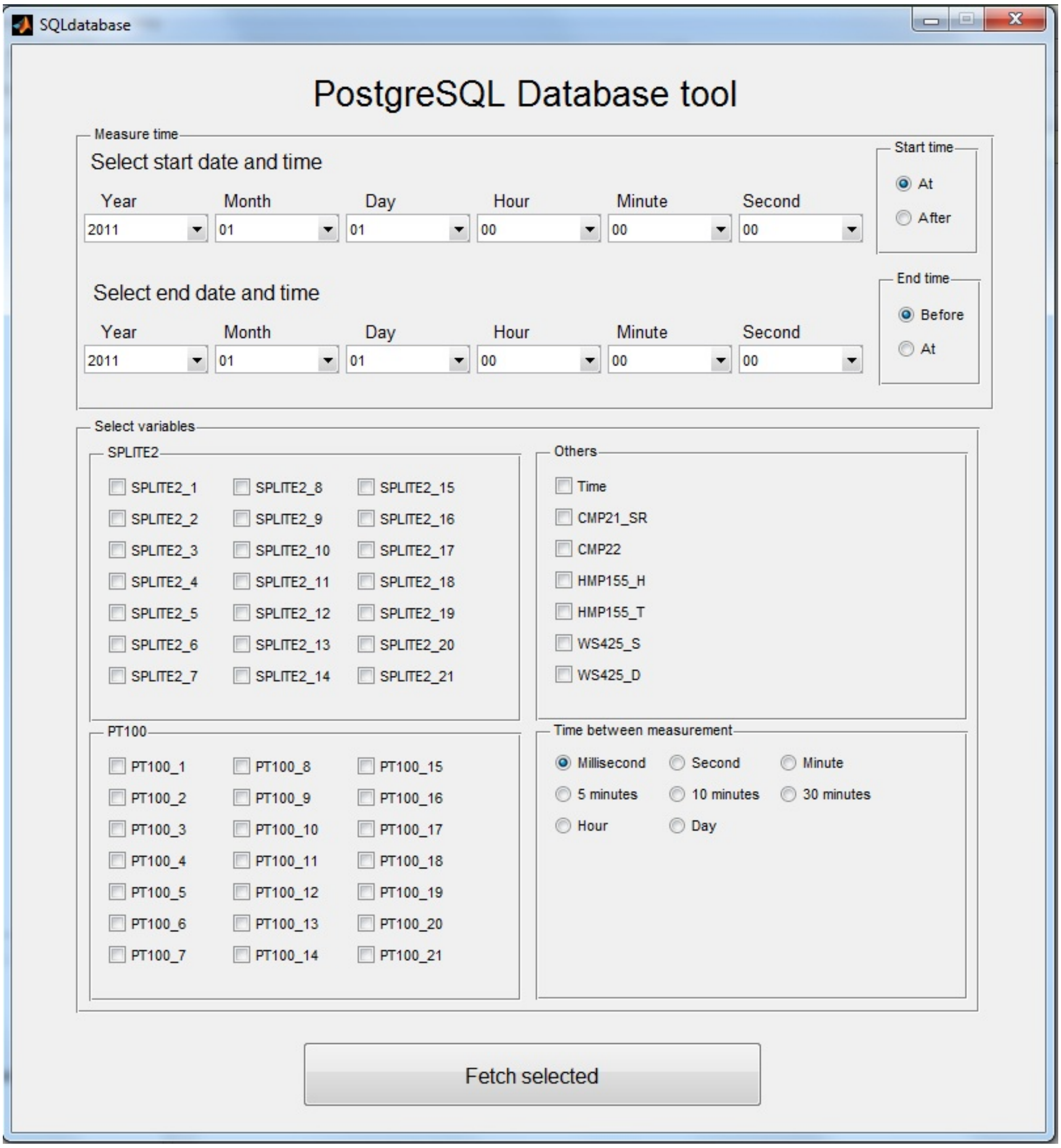

Figure 4.5. *Picture of the MATLAB GUI used to communicate with the PostgreSQL database.*

Now the GUI works for the user so that first the user can select the start and end times. After that the user can choose which sensor data is wanted from that time line. Lastly the user chooses the time between the measures and clicks *fetch*, and the rest is handled inside the GUI. The user will see information in MATLAB window regarding the fetch process and after it is completed, a data variable will appear into MATLAB which holds all the fetched data.

This MATLAB GUI holds two special cases written in code that require clarifying. The first one is how the time between measurements is handled and the second is inside limitation.

The time between measurements is handled with the information of data points. We know the data point saving speed, and therefore we know how many points are generated in one second to the database and so on. Now as this number is known, if user selects for example one minute interval between measurements, SQL is formed so that it takes one value and skips so many values as there are generated in a minute based on the data generation speed and does the same over and over again until the end date and time have been reached.

Most unfortunately a limitation had to be added to this GUI. It limits the maximum number of data values received during one fetch. If it would not limit the amount, MATLAB would crash into a Java memory error. What caused this required a lot of studying, and it was high priority because of being so crucial.

The MATLAB database tool box uses cursors to get data from the database. The problem that is causing the error comes from the behavior of how MATLAB handles the cursor with databases, at least with PostgreSQL. Corresponding PostgreSQL JDBC driver manual part is available on the Internet. [28] According to that by default the whole result set is fetched at once which can become an issue with large data sets. There are restrictions which force the driver to fetch the whole result set at once, such as if a connection is made to be in autocommit mode, ResultSet.TYPE_FORWARD_ONLY is not used or the query is not given in single statement. MATLAB's tool box fails in one or many of these and tries to fetch the whole result set every time and then runs into memory problems with Java. Now MATLAB's only solutions for this are increasing Java Heap Size from preferences or just decreasing the amount of data points fetched at one time. Neither of these is a proper solution, the first one would just raise the limit a little before the error would appear and the second one is just limiting.

Something more had to be done regarding this to fix the use of the data. So a research was started about how the same could be done without MATLAB's database toolbox. After a long research, another way was found to do this and as the JDBC driver restrictions were taken into account, it was possible to fetch unlimited amount of data at one time. This was great news, but there was a problem. MATLAB has very optimized language and Java calls are extremely expensive in time in MATLAB. The solution had multiple Java calls and MATLAB's tool box has all Java code inside one call which is why the solution was so slow that it was not practical anymore. It would have been possible to make a complex single Java call like MATLAB's tool box, but that would have meant spending a lot of time to make a part of MATLAB's database tool box, and because it is a paid product, a bug report was made instead. At first the only replies to the bug report were the solutions already known, but after demonstrating the problem more deeply with working a MATLAB script, it was promised to take this matter into development.

4.6. The Internet site for the photovoltaic research power plant

A public Internet site for the system was the last thing that was proposed for the system. It had minor priority but because it seemed like an interesting challenge and it could be used in promotional purposes, it was chosen to be done.

There are many different web languages to make web sites. Because of previous experience and favor of open source, the site was decided to be built around Javascript, Ajax, PHP, xhtml, and CSS languages.

Initially, designing the web page began with researching what would be needed and what would be shown. It was decided that there would be only some general information about weather and the system. To be able to show different days, a way to make a calendar was needed. New HTML5 would have done this, but it is still under development and the support from different browsers is not very good. jQuery and jQueryUI were found, and those seemed perfect for the job as they are free to use with an MIT license. It was also noted that vr.fi uses these two.

A tab-type outfit was made for the webpage and the calendar from jQueryUI was used, both of which were customized. To show graphs from different weather statistics, the CSS hover features were used. The layout is mostly done with CSS, though because of the hover effect some locations had to be defined static. A picture of the web site can be seen in Figure 4.6.

Regarding the coding of the page, the most important aspect is security and soon after that comes behavior in errors. Because the web site is connected to a database, SQL injection attacks are the biggest threat. To protect against this, two different countermeasures were taken. First, input validator was made to make sure that certain inputs are valid. Second, Web page access to the database was limited to only simple things. Building error handling countermeasures is also a difficult task because it is not always known what kind of errors can occur, so countermeasures were taken against obvious errors.

As a future development, improving graphs behavior could improve the overall web experience. For instance, flot for jQuery would allow interactions with the graph data. The current design fetches one graph image (size of a few KB) from database and converts it into an image. Using flot would mean fetching numerous data points from the database at once. There would not probably be a big difference either way in speed.

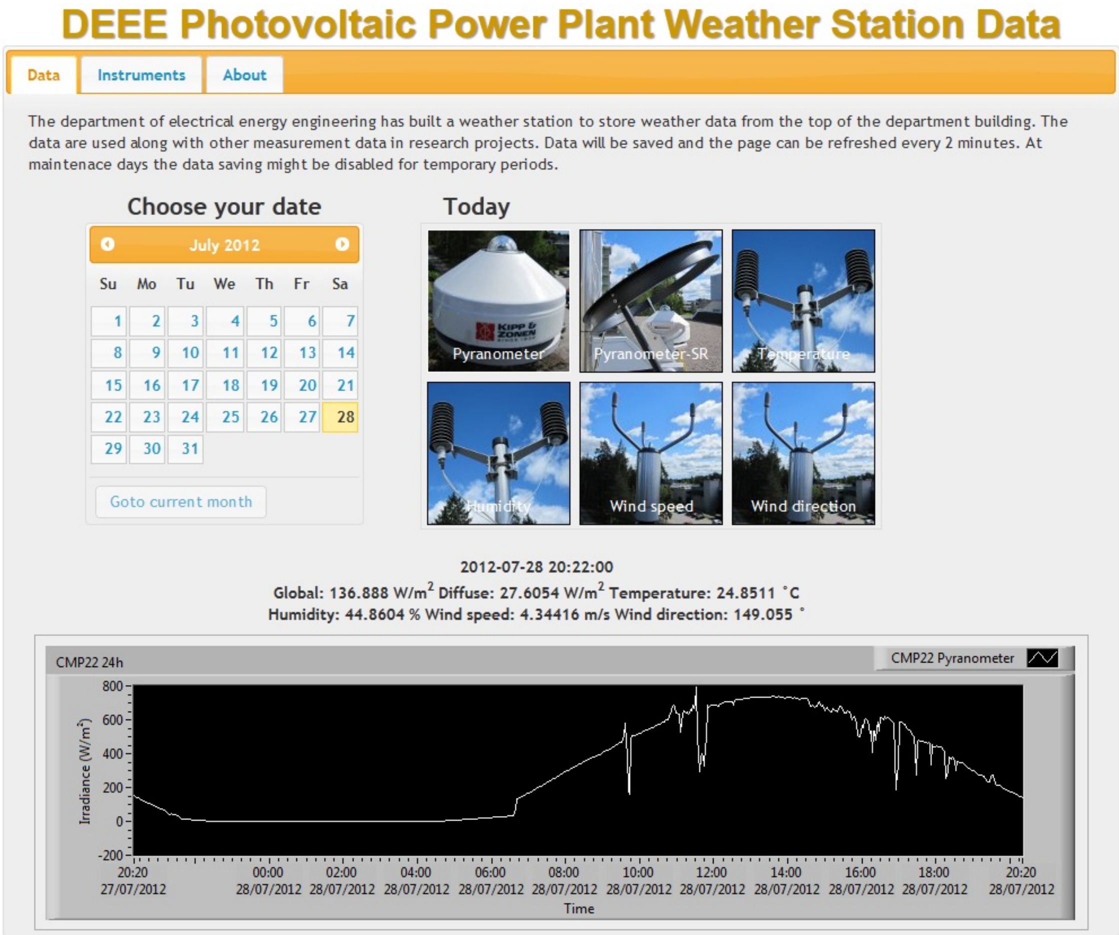

Figure 4.6. Picture of the public website

4.7. The data handling in LabVIEW

LabVIEW is a software from National Instruments. As National Instruments' modules and data logger were already bought, it was known that LabVIEW software would be used as a tool to program modules and data handling. Learning the graphical language in LabVIEW requires one's own study, dedication and effort because it is not taught in TUT. During the system developing, National Instrument staff held one class of the basics on how to use the LabVIEW software. LabVIEW itself holds numerous examples of various aspects and a help guide. With the help from experts and by studying LabVIEW thoroughly, the whole LabVIEW coding was completed.

Each created VI in LabVIEW has its own graphical user interface (GUI). These GUIs hold controls and also description for the VI. With controls the users are able to make changes during VI runtime.

At the very beginning after the hardware had been connected, the next thing to do was to connect it with LabVIEW. This was done with MAX (Measurement & Automation Explorer). With MAX, hardware software was updated, meaning in this case cRIO 9074 hardware software. Updating cRIO 9074 hardware software was required in order to get the latest fixes and additions and also the required features for this system. Primary the required running software for cRIO 9047 had to support scan engine, FPGA and time adjustments. In addition NI-Industrials Communications EtherCAT had to be installed to cRIO 9074. This was needed for the slave rack NI 9144 which allowed the connection between NI 9144 and cRIO 9074. Unfortunately this software is not shared with LabVIEW installation discs so it had to be downloaded from National Instruments' web site.

The next thing to do was to create the LabVIEW project for the system. The changes to the modules are done from the project. For instance, the voltage ranges for voltage module can be changed from there and whether modules are working in scan mode or fpga mode. Both NI 9205 modules were adjusted mostly to the minimum voltage range because most of the sensors sent low voltage, the only exceptions being the wind sensor and the humidity sensor. Because none of the current sensors required superior speeds, all of them were set under the scan mode. The scan mode simplifies data retrieving and low level data related coding. On the project page the programmer can also see all the devices and all the VI files connected to this project. If there are some specific building, compiling and running instruction requirements, those are configured from Build Specifications in project. One important aspect on the project page is the VI share between computer and the cRIO 9074 hardware. Each VI which is located under My computer (host computer) runs in the host computer and each under SolarcRIO (crio9074 data logger) runs there. This is really important to understand because eventually the data has to be brought to the host computer. Therefore it has to be carefully planned what kind of operations are processed on the host computer and hardware. In the end, after the system design and implementation were completed the project page looked like in Figure 4.7.

The first thing to do in the system implementation coding process was to create the data fetching VIs for the data modules inside the hardware. Because of the noise in voltage modules' data, it was decided to improve the data signal by reducing the noise. Each voltage sensor data was averaged with Time Averaging function and the averaged values were collected into a table. Also, each correction factor value was added here. Scan mode loop was used in other VI to put this averaging to run in 200 Hz rate. From there the values were put in another scan mode loop to run at 10 Hz rate which sends the values to the host computer on each loop round. Now what actually is created here is a situation where 20 values from each sensor are collected, averaged and sent to the host computer 10 times a second. The timestamp in each sent averaged value is the last of the averaged values, meaning the 20th value. It was tested if increasing the number of values averaged would increase the precision even more, but, the only major increase was found between 2 and 10 averaged values. This made it pointless to increase the noise reduction with extra measures such as different filters. After all, the remaining noise in signal was aproximately one to two W depending on the total radiation, and it was more likely to have been caused by the atmosphere than signal noise.

Inside the host computer the data downloading and handling VI runs a loop with 10 ms speed (100 Hz) that compares the last arrived data time stamp to the last one. This is how it recognizes when new data arrives and does not overload the whole VI's every loop if there is no new data. The reason why the same speed as the sending 10 Hz could not be set is that there are unexpected delays which could cause some data points to be lost. All the data from the sensors and the cRIO 9074 hardware comes in custom made two-dimensional table. The reason for creating such a table was to make sure that each time data is received we receive data value points from all sensors.

As for the first inspection, graphs were created for the data to the host computer VI. There are graphs for each type of sensor to show few minutes data and the averaged 24-hour data. This made it possible to monitor if everything was right with the sensors and can be taken advantage later on.

Other than the graphs in the main VI, the host computer has three big coding parts. The first is saving the data into binary files. The second is saving the data into the database. The last is saving the 24-hour graph images into the database for web site uses. Before any of these were developed and implemented, the current way of the initial data fetching and handling was locked. With the assumption that it would not be altered later on. This happened at the beginning of March 2011.

Saving data into binary files is quite straightforward in LabVIEW. First you open or create and open the file, then write to the file and last close the file. Now where it gets complicated is that usually these kinds of file saving processes are done in LabVIEW by running the VI until it finishes copying and stops. The current system is a real time

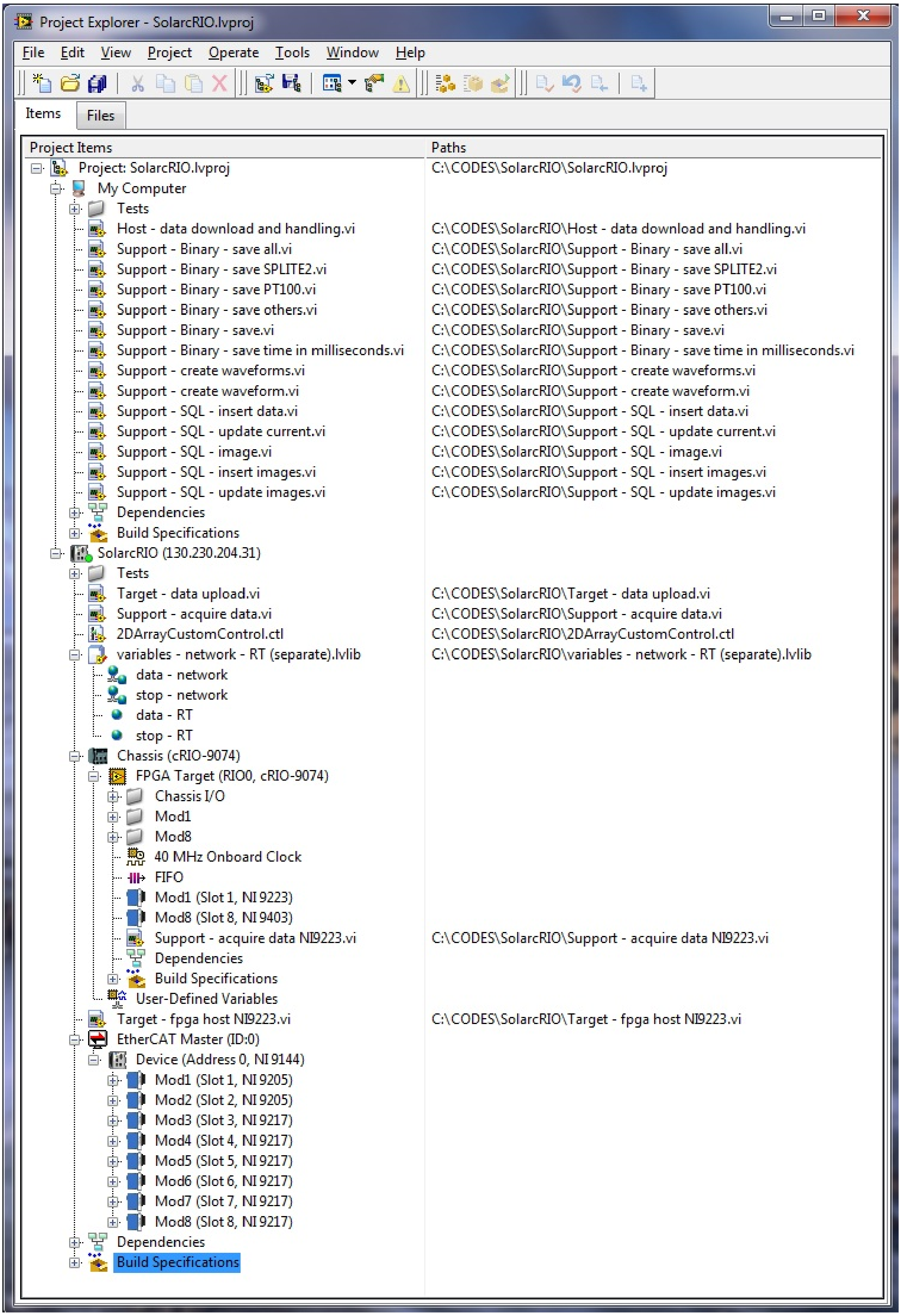

Figure 4.7. *Picture of the system project in LabVIEW after everything was completed*

system which means that the host computer is receiving data all the time, meaning that it runs continuously and does not stop unless stopped or an error occurs. This is why it had to be created more complicated than it is in normal cases. A button was created in the dedicated host computer main VI GUI which can be turned on and off during runtime and the system starts and stops the saving in binary files. If any error occurs in the system, it also sends a message to the binary saving code part to stop. Inside the binary save there is a complicated reference method which recognizes if the arriving value is saved into an existing binary file or if a new one is needed to be created and also when files are opened and closed. By using this complicated system computer resources are saved: the file is not opened or closed every time when values are saved, only when they require opening and closing. By default if an VI is cloned in LabVIEW, it is run so that each close is shared to decrease memory usage. Now with the current reference method this would have caused a huge problem and so it was changed that each cloned reference VI ran as an independent VI.

LabVIEW offers a database toolkit for different database related activities. To test different databases, they were populated with LabVIEW. Three different ways to populate a database from LabVIEW were tested. The first method was to add a row of data with a single and simple string commit. The second method used LabVIEW's own optimization regarding the saving part of the command and not creating it completely with each commit. The third, last and fastest method added multiple rows of single strings with each commit. The test database table was populated with one year of random data in a correct form.

In the host computer main VI the database data uploading was done in trivial string commit LabVIEW method. It commits each data row independently so there would not be any data loss if an error occurred. In terms of the database data adding speed this method did not fall behind with LabVIEW's second method mentioned in last paragraph. Database populating was also protected against common errors, such as errors that occur elsewhere. If not prepared for these, there might occur unexpected problems in the database.

The last large effort in LabVIEW coding was required for uploading LabVIEW chart images into the database. This is done so that once every two minutes, pictures from the 24-hour charts are saved to the host computer. Then these images of few kB (kilo Bytes) in size are read and converted into a hex string. Finally this hex string is uploaded to the database. An image of this in LabVIEW is shown in Figure 4.8. Saving these images in hex format was the best option in a PostgreSQL database because it means that the data is saved directly to the database table and not the links to the images. Along with these images, every two minutes current values are updated to a current table of sensor values. These are also used for the web site.

Error handling in LabVIEW was given a lot of attention. By current design, if an error

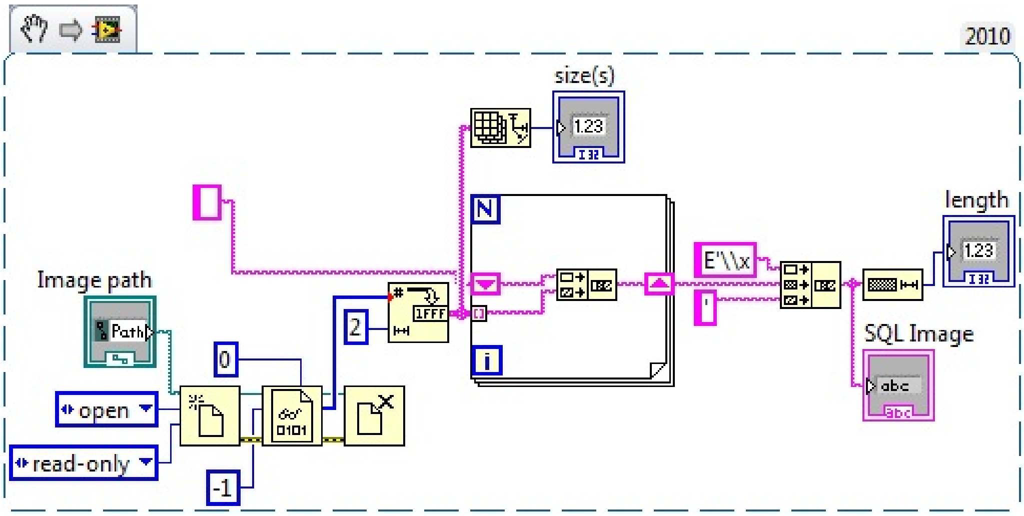

Figure 4.8. *Picture of the database image converting to hex and adding this value into string and ready to be saved into PostgreSQL database*

occurs in the data fetching in cRIO 9074 hardware, it stops and sends the error to the main host computer VI. In there it stops the binary file saving and all database saving and then stops. Also, if any kind of critical error happens in main host computer VI, an error is also sent to the hardware VI to stop. It is also configured in a way that if the main host computer VI is manually stopped from running, the hardware VIs also stop.

These LabVIEW softwares were adjusted for specific compiling and running. Hardware VIs are compiled and sent to the hardware so that once the hardware boots, it automatically starts running the hardware VIs. This means that if the hardware VIs have been stopped, an easy way to make them run again is to reboot the cRIO 9074 hardware. The host computer VIs compiling has been made so that an executable program is created. This executable program is set to start running after the computer starts automatically.

After all these important and complicated codes the main host computer VI would have been crowded and hard to read if additional VIs had not been created. These additional VIs hold most of the data that was not required to be seen directly in the main host computer VI. An icon for each additional VI was created which had text, image or both to help the identification. In general, each VI has a short description written and additional texts where necessary as a helpful documentation. The total number of VIs running all the time is 17, and 33 additional VIs were created for testing purpose.

The last work created in LabVIEW was the initial FPGA data downloading for the new NI 9223 module card. This simply downloads data from the module and outputs it into a graph. It is meant to be extended into other purposes from there.

5. CONCLUSIONS

The data acquisition and storing system for the photovoltaic research power plant is fully working and running. It consists of several different parts, such as different sensors, connections, grounding, file-based system, database, MATLAB, Internet site and LabVIEW.

The system has 21 SP Lite 2 and Pt100 sensors to measure radiation and temperature. There are two weather stations consisting of HMP155 humidity and temperature sensor, CMP22 pyranometer, CMP21 pyranometer with shadow ring and WS425 wind sensor.

All these are connected into NI 9144 expansion chassis via 2 NI 9205 modules and 6 NI 9217 modules. Each sensor cable is extended with an extension cable. The weather station devices are extended once in the second connection room while the rest have direct connection to the modules in the first connection room.

The grounding includes a thunder rod for protection and different grounding choices. The SP Lite 2 sensors are grounded from structure and other sensors are grounded from the shield in the first connection room.

Database system is the primary method to save the data. File-based system offers an alternative method to save the data and process it. It is the fastest and most compressed method. The database management system offers many advantages compared to file-based system. This can be improved in the future as the usage is better known.

MATLAB was the primary analytic tool to analyze the data. A custom GUI has been created for MATLAB in order to make it easier to get the data from the database without a need to learn SQL. This is also one point where more improvements can be made in case of new needs.

The Internet site for this system offers general data to public. This can easily be extended in the future based on needs.

LabVIEW handles all the logic and data processing and saving methods. In other words it is the core of the system. Based on the needs and future additions, it remains the main place to expand the system and therefore understanding it is critical.

The information above summarizes the work that was done to create the measuring and data storing system for the photovoltaic research power plant. In the course of the process of developing the system, different alternatives of how to execute different stages of the system were carefully considered. Now that the system is finished and fully working, it is safe to say that the whole process as well as the outcome were successful.

REFERENCES

[1] Luque, A. & Hegedus, S. Handbook of Photovoltaic Science and Engineering. 2003, John Wiley & Sons Ltd. 1115 p.

[2] Lynn, P.A. Electricity from Sunlight: An Introduction to Photovoltaics. 2010, John Wiley & Sons Ltd. 221 p.

[3] Solar Book. On-Line Solar Study Reference. 2011. [accessed on 20.6.2011]. Available at: http://www.solarbook.ie/solar-panel-physics.html.

[4] Tanenbaum A.S. & Van Steen, M. Distributed Systems Principles and Paradigms. 2nd edition. 2007, Pearson Education. 686 p.

[5] Connolly, T. & Begg, C.E. Database Systems A Practical Approach to Design, Implementation, and Management. 4th edition. 2005, Pearson Education Limited. 1374 p.

[6] Lobera, D.T. Measuring actual operating conditions of a photovoltaic power generator. Dissertation. Tampere 2010. Tampere University of Technology. 103 p.

[7] Campbell Scientific (Canada) Corp. Instruction Manual SP Lite2 Silicon Pyranometer. 2010. [accessed on 17.6.2011]. Available at: http://www.campbellsci.ca/Catalogue/SPLite2.html.

[8] KIPP & ZONEN. KIPP & ZONEN CMA series and CMP series Instruction Manual. 2010. [accessed on 18.6.2011]. Available at: http://www.kippzonen.com/?product/15112/CMP+22.aspx.

[9] KIPP & ZONEN. KIPP & ZONEN CM 121 Shadow Ring Instruction Manual. 2004. [accessed on 18.6.2011]. Available at: http://www.kippzonen.com/?product/4281/CM+121B+C+Shadow+Ring.aspx.

[10] KIPP & ZONEN. KIPP & ZONEN CVF 3 Ventilation Unit. 2009. [accessed on 18.6.2011]. Available at: http://www.kippzonen.com/?product/7071/CVF+3+Ventilation+Unit.aspx.

[11] KIPP & ZONEN. KIPP & ZONEN 2011 Product Catalogue Solar Instruments, Atmospheric Science Instruments and Wind Sensors. 2011. [accessed on 19.6.2011]. Available at: http://www.kippzonen.com/?product/38192/CMF+Mounting+plates.aspx.

[12] VAISALA. User's guide Vaisala WINDCAP Ultrasonic Wind Sensor WS425. 2008. [accessed on 18.6.2011]. Available at: http://www.vaisala.fi/instrumentit/tuotteet/ws425.html.

[13] VAISALA. User's guide Vaisala HUMICAP Humidity and Temperature Probe HMP155. 2008. [accessed on 18.6.2011]. Available at: http://www.vaisala.fi/instrumentit/tuotteet/hm-hmp155.html.

[14] VAISALA. DTR500 Solar Radiation and Precipitation Shields. 2008. [accessed on 18.6.2011]. Available at: http://www.vaisala.fi/weather/products/dtr500.html.

[15] National Instrument. CompactRIO Integrated Systems with Real-Time Controller and Reconfigurable Chassis NI cRIO-907x. 2011. [accessed on 19.5.2012]. Available at: http://sine.ni.com/ds/app/doc/p/id/ds-204/lang/en.

[16] National Instrument. Deterministic Ethernet Expansion Chassis. 2012. [accessed on 19.5.2012]. Available at: http://www.ni.com/pdf/products/us/cat_ni9144.pdf.

[17] National Instrument. NI 9217. 2010. [accessed on 19.5.2012]. Available at: http://sine.ni.com/ds/app/doc/p/id/ds-193/lang/en.

[18] National Instrument. NI 9205. 2010. [accessed on 19.5.2012]. Available at: http://sine.ni.com/ds/app/doc/p/id/ds-190/lang/en.

[19] National Instrument. Labview. 2011. [accessed on 20.5.2012]. Available at: http://www.ni.com/labview/.

[20] The PostgreSQL Global Development Group. About. 2012. [accessed on 20.5.2012]. Available at: http://www.postgresql.org/about/.

[21] MathWorks. MATLAB The Language of Technical Computing. 2011. [accessed on 20.5.2012]. Available at: http://www.mathworks.se/tagteam/70533_91199v01_MATLABDataSheet_v9.pdf?s_cid=ML2012_bb_datasheet.

[22] The Apache Software Foundation. What is Apache?. 2012. [accessed on 20.5.2012]. Available at: wiki.apache.org/httpd/.

[23] The Apache Software Foundation. I'm not a lawyer. What does it all MEAN?. 2012. [accessed on 20.5.2012]. Available at: wiki.apache.org/httpd/.

[24] Achour, M., Betz, F., Dovgal, A., Lopes, N., Magnusson, H., Richter, G., Seguy, D. & Vrana, J. PHP Manual. 2012, The PHP Group. [accessed on 20.5.2012]. Available at: http://www.php.net/manual/en/.

[25] The jQuery Foundation. jQuery: The Write Less, Do More, JavaScript Library. 2012. [accessed on 20.5.2012]. Available at: http://jquery.com/.

[26] jQuery Foundation and the jQuery UI Team. jQuery user interface. 2012. [accessed on 20.5.2012]. Available at: http://jqueryui.com/.

[27] Kumpula, M. Kattilaitoksen ulkoinen ja sisäinen ukkossuojaus. Dissertation. Tampere 2007. Tampere University of Technology.

[28] PostgreSQL Global Development Group. Getting results based on a cursor. 2011. [accessed on 20.5.2012]. Available at: http://jdbc.postgresql.org/documentation/91/.

APPENDIX 1: PHOTOVOLTAIC RESEARCH POWER PLANT LAYOUT

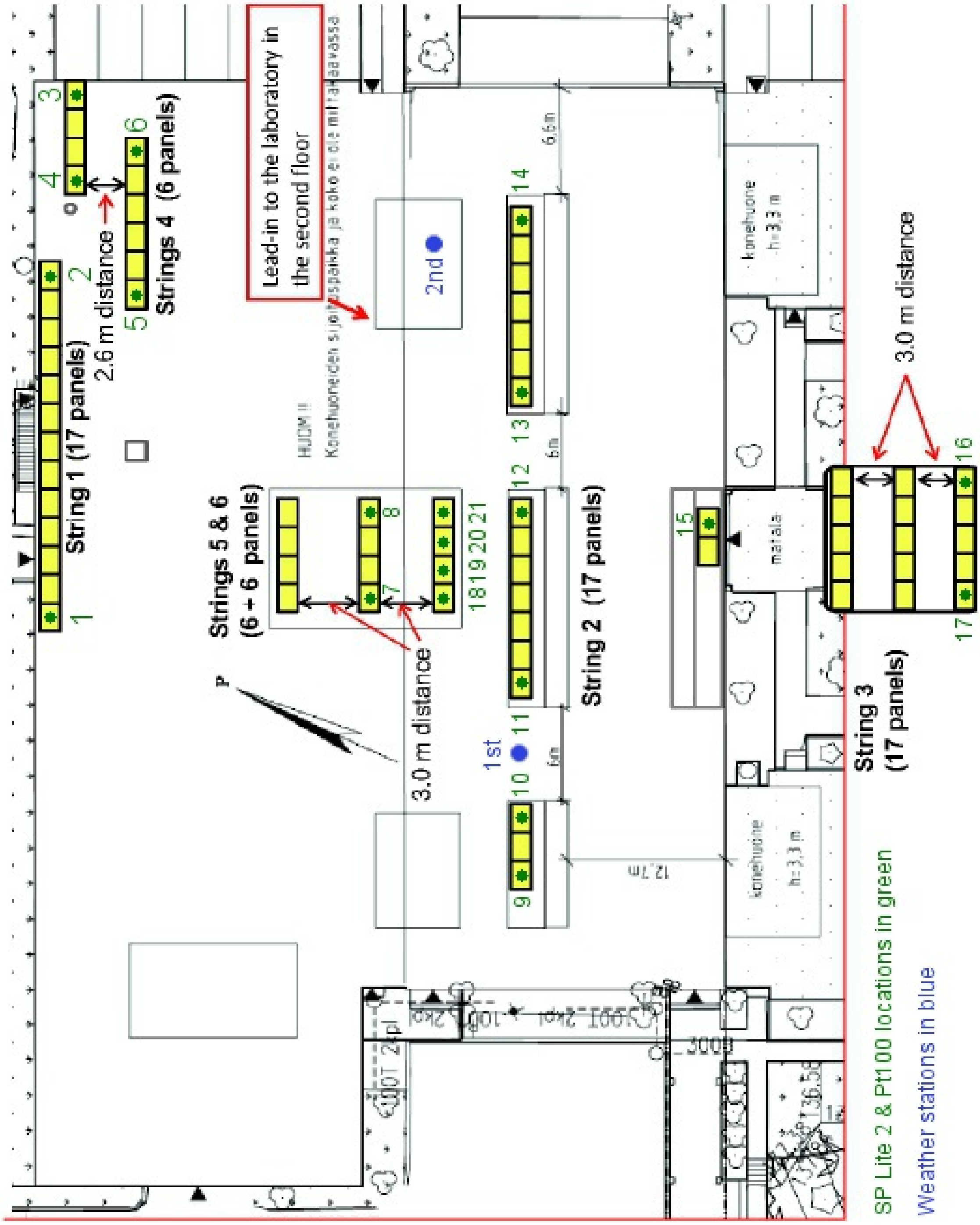

Figure appendix 1.1. Photovoltaic research power plant layout

APPENDIX 2: THE ELECTRICAL CONNECTIONS OF PHOTOVOLTAIC RESEARCH POWER PLANT STRINGS

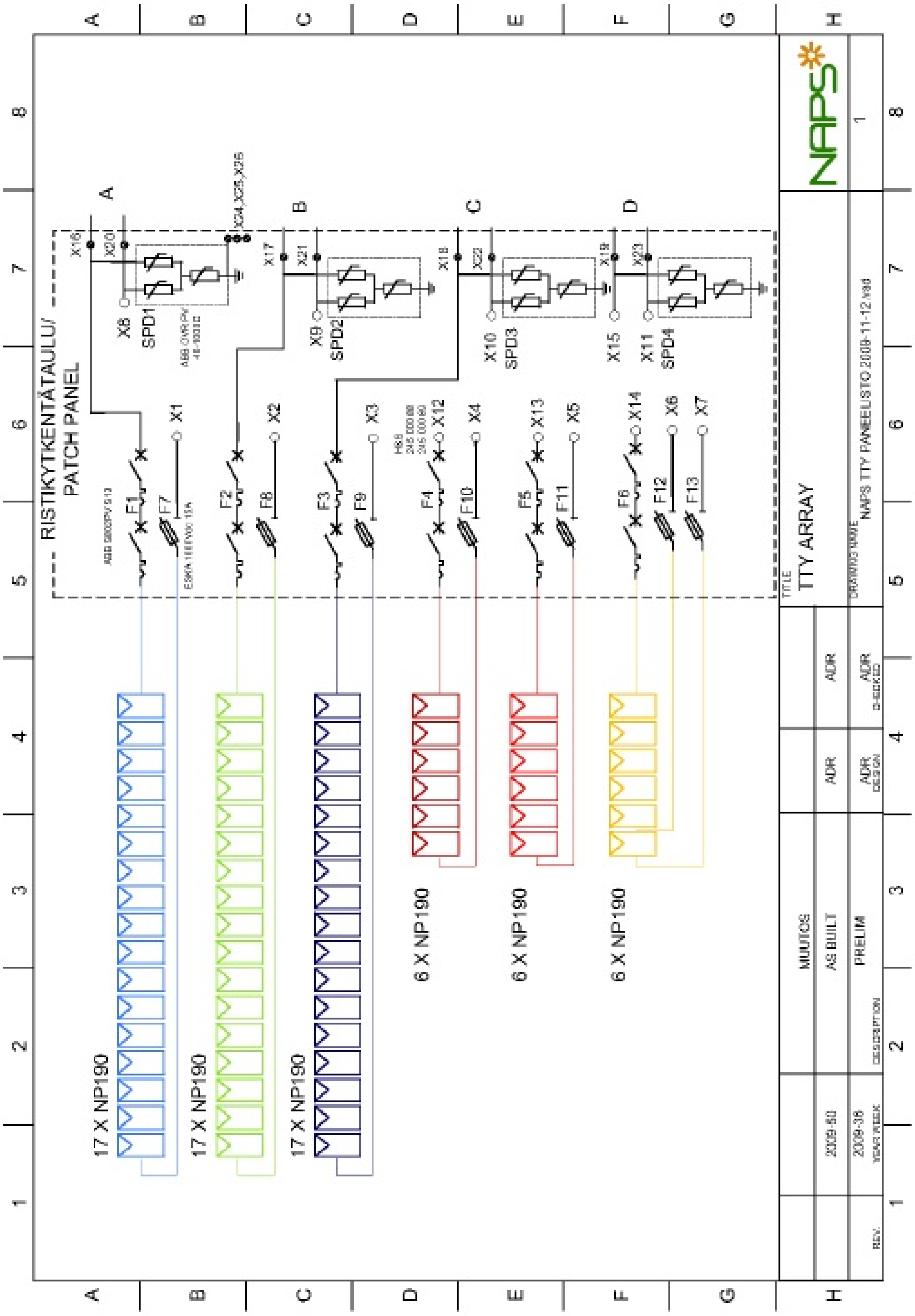

Figure appendix 2.1. Photovoltaic research power plant strings' electrical connections

APPENDIX 3: THE CONNECTION SCHEME OF PHOTOVOLTAIC RESEARCH POWER PLANT STRINGS

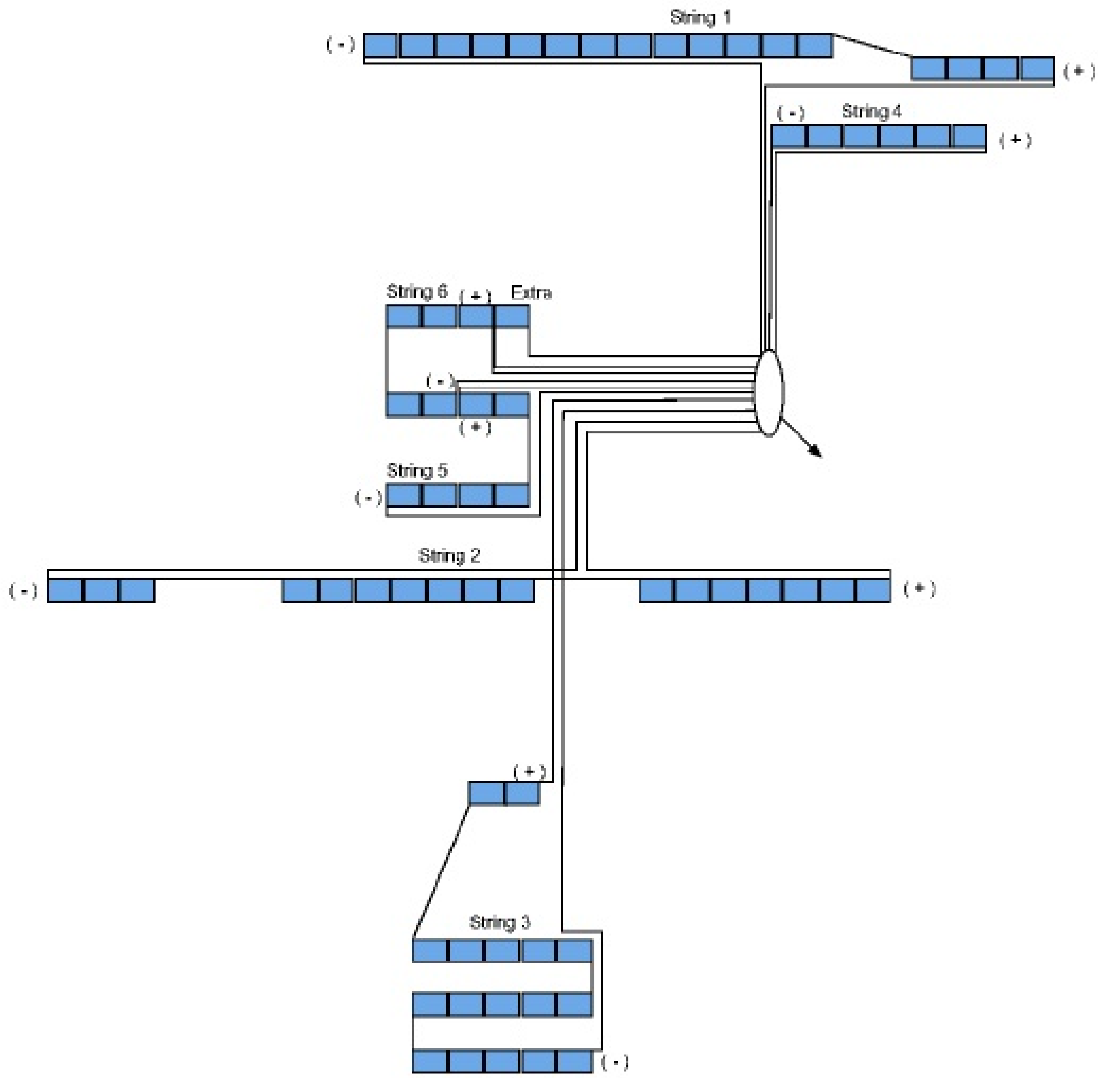

Figure appendix 3.1. Photovoltaic research power plant strings' connection scheme